

Universidad de las Ciencias Informáticas
Facultad 1



Título: Módulo de Visitas para el sistema de Control de Acceso

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autoras: Irina Fuentes Viñas

Eilyn Guzmán Rodríguez

Tutor: Ing. Ariel Ramírez Álvarez

Lic. Rosmel Álvarez Monzón

Junio 2009

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

Módulo de Visitas para el Sistema de Control de Acceso_y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Irina Fuentes Viñas

Eilyn Guzmán Rodríguez

Ing. Ariel Ramírez
Álvarez

Lic. Rosmel Álvarez
Monzón

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Módulo de Visitas para el Sistema de Control de Acceso.

Autoras: Eilyn Guzmán Rodríguez

Irina Fuentes Viñas

Eifyn

A las personas más importantes en mi vida, quienes de una forma u otra me han permitido llegar hasta aquí y ser la persona que soy:

A mi mamá, por ser la amiga que siempre me ha brindado lo mejor de sí, por ser la estrella que ha iluminado mi camino, por ser la fuerza y ejemplo que necesité para llegar al lugar donde hoy me encuentro.

A mi hermana, a tata, por ser mi compañera, mi apoyo, la persona que siempre ha estado y estará a mi lado. A su esposo Dimitry, por toda su ayuda.

A mi papá, porque a pesar de la lejanía a sabido ser un buen padre, y hacernos sentir que siempre podremos tener su apoyo, al igual que Rosita. A ambos, gracias.

A mi tía Mae y mi primi Ross, que a pesar de la amplia distancia y los pocos encuentros, han seguido paso a paso y detalle a detalle, todo lo ocurrido en estos últimos meses. A mi primo Albert.

A la Yiti por ser mi compañera de alegrías y lágrimas durante los 5 años de la Universidad, gracias por tu compañía.

A mi amiga Maria, por seguir cada avance y obstáculo encontrado en el camino para la confección de este trabajo, gracias por tu apoyo.

A mis amigas Dairy, Yany, Dahily, Adiely y Aniledy por ser las mejores amigas del mundo, las que han sabido serlo desde la infancia y que lo serán toda la vida, no importa que tan lejos estén.

A mi querida vecina Amparo, por su ayuda desde la casa, por cuidar de mi mamá en mi ausencia.

Y a todas las demás personas que de una forma u otra colaboraron en la confección de este trabajo... Gracias a todas...

Deseo ofrecer un agradecimiento especial para Guille, por haberme dado una razón para querer ser cada día mejor.

Irina

Agradecer a mi mamá que la quiero con la vida, a ella que me ha formado, atenta todo el tiempo, protegiéndome, guiándome y educándome.

A mi papá por estar siempre pendiente, por su alegría, por su ejemplo, por la educación. Te quiero mucho

A mis hermanos que los adoro, Dany, Bia y Magui.

A mi abuela, siempre conmigo, por quererme tanto, defenderme y estar siempre que la necesito

A mi abuelo Pite por darme tanto amor, ejemplo en todo momento, yo siempre tan orgullosa de ti. Gracias por todo lo que me enseñaste.

A mi abuela Tita, este es mi regalo.

A mi abuelo René por cuidarme y mimarme tanto.

A mi tío Frank, por animarme cuando estoy triste y estar siempre atento de su hermanita más chiquita.

A Tony, sé que siempre que estés cerca no voy a tocar el suelo...

A Miry y Sonia las mejores tías del mundo, siempre cariñosas, bellas, llenas de consejos.

A mis primos, Juanqui, Naty, Dayi, Jeny, Irko, Rosa M, Brandon, Brian y Cristian por tantos momentos de alegría.

A Sergi, siempre he podido contar contigo, cuando hay alegría y cuando mi mundo está de cabeza, ahí estas tu.

A Daniel y Betty por su apoyo.

A mis amigas, Vallejo, Yulexy, Acrafys, Miry gracias por estar...muy especial a la Yusma y Yanier que me han acompañado todos estos años y me han apoyado incondicionalmente... los quiero mucho.

A Pepe, Joe, Mary y Damian por la ayuda y la paciencia...

A mis suegros por estar pendientes.

A la riquera de mi vida, por tantas y tantas noches, por tu cariño, por apoyarme, por animarme, por quererme, por quererte... como yo te quiero muju....

DEDICATORIA

Eifyn

Dedico mi tesis de diploma como Ingeniera en Ciencias Informáticas a Mi Mamá. Este es mi sueño y es el tuyo también, tú has sido la mayor responsable de que fuera realidad, y esta es mi forma de agradecértelo. Gracias por darme la vida, gracias por estar en ella.

Irina

*A mis padres, mis abuelos, mis tíos y mis hermanos... especialmente para ti mami, que he tenido pocas formas de demostrarte lo mucho que te quiero y lo muy importante que eres para mi, te regalo estos 5 años para que estés tan orgullosa de mi como yo de ti.
A mi abuelo Pite, de tu futura Ingeniera Fuentes.*



No basta saber, se debe también aplicar. No es suficiente querer, se debe también hacer.

Johann Wolfgang Goethe (1749-1832)

RESUMEN

La Universidad de las Ciencias Informáticas (UCI) cuenta con una gran cantidad de recursos, debido a esto es de suma importancia que el acceso de personal ajeno sea precedido por un proceso de solicitud y aprobación. Actualmente este proceso se efectúa mediante la petición personal de un pase impreso, como aval de autorizo de entrada.

Este procedimiento tiene en contra dos factores de importancia: la disponibilidad de directivos y la seguridad, ya que los directivos no siempre se encuentran disponibles y la autorización impresa es sencilla de falsificar.

El Módulo de Visitas para el sistema de Control de Acceso realiza el análisis, diseño e implementación de una aplicación web que permite simplificar los procesos de solicitud y autorización de visitas y controlar la entrada y salida de las mismas mediante un sistema multiplataforma basado en software libre.

Con esta propuesta, resultado de la investigación de los procesos del negocio, posibles herramientas, metodologías y sistemas vinculados al objeto de estudio, se dará fin a los problemas de disponibilidad y seguridad existentes, ya que el sistema permitirá realizar la gestión de solicitudes y autorizaciones desde la red en cualquier zona de la universidad, registrará a todos los visitantes que accedan a la Universidad y brindará reportes de importancia para los distintos usuarios.

Palabras claves: acceso, visita, solicitud, autorización.

INDICE

<i>INTRODUCCIÓN</i>	1
<i>CAPITULO 1 FUNDAMENTACIÓN TEÓRICA</i>	6
1.1 Introducción	6
1.2 Necesidad del Módulo de Visitas para el sistema Control de Acceso	6
1.3 Sistemas de Reservación de Visitas	6
1.3.1 Los sistemas de reservación de visitas en el ámbito nacional	9
1.3.2 Los sistemas de reservación de visitas en la UCI	10
1.3.3 Motivos de su no selección	11
1.4 La aplicación	11
1.5 Caracterización de las herramientas	12
1.5.1 Lenguaje de programación	13
1.5.2 Plataforma de desarrollo	13
1.5.3 Sistema Gestor de Base de Datos	14
1.5.4 Entorno integrado de desarrollo	16
1.5.5 Servidor Web	16
1.5.6 Lenguaje Unificado de Modelado	18
1.5.7 Herramienta CASE	18
1.5.8 Fundamentación de la metodología	19
1.6 Propuesta de solución	23
1.7 Conclusiones	23
<i>CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA</i>	24
2.1 Introducción	24

2.2	Reglas del Negocio	24
2.3	Modelo de negocio	25
2.4	Descripción textual de los casos de uso del negocio	27
2.5	Diagramas de actividades	32
2.6	Modelo de objetos del negocio	32
2.7	Relación de los requerimientos	32
2.8	Modelo de casos de uso del sistema	43
2.9	Estimación del esfuerzo	44
2.9.1	Planificación por Casos de Uso	44
2.9.2	Paso 1 Cálculo de Puntos de Casos de Uso sin Ajustar	45
2.9.3	Factor de Peso de Actores sin ajustar	45
2.9.4	Factor de Peso de Casos de Uso sin Ajustar	46
2.9.5	Paso 2. Cálculo de los puntos de casos de uso ajustados	47
2.9.6	Paso3. Estimación de esfuerzo a través de puntos de casos de uso	49
2.9.7	Paso 4. Calcular el esfuerzo de todo el proyecto	50
2.9.8	Costo del proyecto	51
2.10	Beneficios tangibles e intangibles	52
2.11	Análisis de Costo y Beneficios	52
2.12	Conclusiones	52
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA		54
3.1	Introducción	54
3.2	Análisis	54
3.2.1	Diagrama de clases del análisis	54

3.3	Diseño	56
3.3.1	Diagrama de clases web del diseño	57
3.3.2	Descripción de las clases web del diseño	60
3.3.3	Diagramas de interacción	60
3.4	Patrones de diseño y de arquitectura que implementa Symfony	62
3.4.1	Patrones aplicados en el diseño	62
3.4.2	Otros patrones aplicados en el diseño	63
3.4.3	Utilización del patrón que implementa Symfony	64
3.5	Diseño de la base de datos	66
3.5.1	Modelo entidad relación de la base de datos	66
3.5.2	Descripción de las tablas de la base de datos	67
3.6	Tratamiento de errores	74
3.7	Seguridad	74
3.8	Principios del diseño	75
3.9	Conclusiones	75
<i>CAPITULO 4 Implementación y Prueba</i>		76
4.1	Introducción	76
4.2	Implementación	76
4.3	Diagramas de Componentes	78
4.4	Diagrama de Despliegue	82
4.5	Pruebas	82
4.5.1	Métodos de Prueba	83
4.5.2	Prueba del Sistema	83

4.6 Conclusiones _____	91
<i>CONCLUSIONES GENERALES</i> _____	92
<i>RECOMENDACIONES</i> _____	93
<i>REFERENCIAS BIBLIOGRAFICAS</i> _____	94
<i>BIBLIOGRAFIA</i> _____	96
<i>GLOSARIO DE TÉRMINOS Y SIGLAS</i> _____	98
<i>ANEXOS</i> _____	100

INDICE DE TABLAS

<i>Tabla 2.1 Descripción de actores del negocio.</i>	26
<i>Tabla 2.2 Descripción de trabajadores del negocio.</i>	27
<i>Tabla 2.3 Descripción textual del CU Solicitud de visita diurna.</i>	29
<i>Tabla 2.4 Descripción textual del CU Solicitar visita temporal.</i>	30
<i>Tabla 2.5 Descripción textual del CU Solicitar visita al hotelito.</i>	32
<i>Tabla 2.6 Descripción de actores del sistema.</i>	43
<i>Tabla 2.7 Factor de Peso de los Actores sin Ajustar.</i>	46
<i>Tabla 2.8 Factor de Peso de Casos de Uso sin Ajustar.</i>	46
<i>Tabla 2.9 Factor de complejidad técnica.</i>	48
<i>Tabla 2.10 Factor del Factor Ambiente</i>	49
<i>Tabla 2.11 Esfuerzo en Horas-Hombre.</i>	50
<i>Tabla 3.1 Descripción de la tabla: estado_de_visita.</i>	68
<i>Tabla 3.2 Descripción de la tabla: estado de visitante.</i>	68
<i>Tabla 3.3 Descripción de la tabla: área.</i>	68
<i>Tabla 3.4 Descripción de la tabla: organismo.</i>	69
<i>Tabla 3.5 Descripción de la tabla: visita</i>	69
<i>Tabla 3.6 Descripción de la tabla: categoría_de_solicitud.</i>	69
<i>Tabla 3.6 Descripción de la tabla: tipo_solicitud.</i>	70
<i>Tabla 3.7 Descripción de la tabla: solicitud.</i>	71
<i>Tabla 3.8 Descripción de la tabla: atributo_de_solicitud.</i>	71
<i>Tabla 3.9 Descripción de la tabla: visitante.</i>	71
<i>Tabla 3.10 Descripción de la tabla: directivo.</i>	72
<i>Tabla 3.11 Descripción de la tabla: solicitante</i>	72

<i>Tabla 3.12 Descripción de la tabla: valor_atributo_visitante.</i>	72
<i>Tabla 3.13 Descripción de la tabla: tipo_de_visitante.</i>	73
<i>Tabla 3.14 Descripción de la tabla solicitud_visitante.</i>	73
<i>Tabla 3.15 Descripción de la tabla: valor_atributo_visitante.</i>	73
<i>Tabla 3.16 Descripción de la tabla: valor_atributo_de_solicitud.</i>	74
<i>Tabla 3.17 Descripción de la tabla: estado_de_solicitud.</i>	74
<i>Tabla 4.1 CP1. CU Gestionar Solicitud Diurna. Sección: Nueva Solicitud Diurna</i>	84
<i>Tabla 4.2 CP1. CU Gestionar Solicitud Diurna. Sección: Modificar Solicitud Diurna</i>	85
<i>Tabla 4.3 CP1. CU Gestionar Solicitud Diurna. Sección: Cancelar Solicitud Diurna</i>	85
<i>Tabla 4.4 CP1. CU Gestionar Solicitud Diurna. Sección: Detalles Solicitud Diurna</i>	86
<i>Tabla 4.5 CP2. CU Gestionar Solicitud Temporal. Sección: Nueva Solicitud Temporal.</i>	86
<i>Tabla 4.6 CP2. CU Gestionar Solicitud Temporal. Sección: Modificar Solicitud Temporal.</i>	87
<i>Tabla 4.7 CP2. CU Gestionar Solicitud Temporal. Sección: Cancelar Solicitud Temporal.</i>	87
<i>Tabla 4.8 CP2. CU Gestionar Solicitud Temporal. Sección: Detalles Solicitud Temporal.</i>	87
<i>Tabla 4.9 CP3. CU Gestionar Solicitud Hotelito. Sección: Nueva Solicitud Hotelito.</i>	88
<i>Tabla 4.10 CP3. CU Gestionar Solicitud Hotelito. Sección: Modificar Solicitud Hotelito.</i>	88
<i>Tabla 4.11 CP3. CU Gestionar Solicitud Hotelito. Sección: Cancelar Solicitud Hotelito.</i>	89
<i>Tabla 4.12 CP3. CU Gestionar Solicitud Hotelito. Sección: Detalles Solicitud Hotelito.</i>	89
<i>Tabla 4.13 CP4. CU Aprobar/Denegar Solicitudes.</i>	89
<i>Tabla 4.14 CP5. CU Registrar Entrada/Salida. Sección Nueva Visita</i>	90
<i>Tabla 4.15 CP5. CU Registrar Entrada/Salida. Sección Registrar Entrada/Salida de Visita</i>	90

INDICE DE FIGURAS

<i>Figura 1 Análisis realizado por Netcraft de la explotación de Servidores Web</i>	17
<i>Figura 2 Porcentaje de utilización de las diferentes metodologías.</i>	20
<i>Figura 3 Ciclo de retroalimentación y procesos de desarrollo XP.</i>	21
<i>Figura 4 Diagrama de casos de uso del negocio.</i>	27
<i>Figura 8 Diagrama de Casos de Uso del Sistema.</i>	44
<i>Figura 9 Diagrama de clases del análisis CU: Gestionar visita diurna.</i>	55
<i>Figura 10 Diagrama de clases del análisis CU: Gestionar visita temporal.</i>	56
<i>Figura 11 Diagrama de clases del análisis CU: Gestionar visita al hotelito.</i>	56
<i>Figura 12 Diagrama de clases del análisis CU: Aprobar/Denegar solicitudes.</i>	56
<i>Figura 13 Diagrama de clases del análisis CU: Registrar entrada/salida de visitantes.</i>	56
<i>Figura 14 Diagrama de clases del diseño del CU Gestionar visita diurna.</i>	57
<i>Figura 15 Diagrama de clases del diseño del CU Gestionar visita temporal.</i>	58
<i>Figura 16 Diagrama de clases del diseño del CU Gestionar visita al hotelito.</i>	58
<i>Figura 17 Diagrama de clases del diseño del CU Aprobar/denegar solicitudes.</i>	59
<i>Figura 18 Diagrama de clases del diseño del CU Registrar entrada/salida de visitantes.</i>	59
<i>Figura 19 Diagrama de secuencia CU Gestionar visita diurna. Escenario: Nueva solicitud diurna.</i>	60
<i>Figura 20 Diagrama de secuencia CU Gestionar visita diurna. Escenario: Modificar solicitud diurna.</i>	61
<i>Figura 21 Diagrama de secuencia CU Gestionar visita diurna. Escenario: Cancelar solicitud diurna.</i>	61
<i>Figura 22 Diagrama de secuencia CU Gestionar visita diurna. Escenario: Detalles solicitud diurna.</i>	62
<i>Figura 23 El patrón MVC.</i>	64
<i>Figura 24 Flujo de trabajo de Symfony.</i>	66
<i>Figura 25 Modelo Entidad Relación.</i>	67
<i>Figura 26 Diagrama de componentes del CU: Gestionar visita diurna.</i>	79

Figura 27 Diagrama de componentes del CU: Gestionar visita temporal. _____ 80

Figura 28 Diagrama de componentes del CU: Gestionar visita al hotelito. _____ 81

Figura 29 Diagrama de componentes del CU: Aprobar/denegar solicitudes. _____ 81

Figura 30 Diagrama de componentes del CU: Registrar entrada/salida de visitantes. _____ 82

*Figura 31 Diagrama de despliegue del sistema.*_____ 82

INTRODUCCIÓN

A partir del año 1996 se comienza a impulsar en Cuba el uso y desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC), impulso que mantuvo su avance y condujo a la creación en el año 2000 del Ministerio de la Informática y las Comunicaciones (MIC) cuyo objetivo primordial es el uso de las TIC en la economía nacional, la sociedad y al servicio del ciudadano para aumentar sistemáticamente la calidad de vida del pueblo cubano. Siguiendo estos objetivos, a mediados del 2002 surge la UCI, universidad de nuevo tipo nacida al calor de la Batalla de Ideas con un novedoso modelo de formación que combina el estudio con la producción y la investigación.

Este centro, es uno de los principales promotores del uso de las TIC en Cuba ya que, cada año ingresa alrededor de 1000 estudiantes de todas las provincias del país, asegurándose así que no exista un rincón donde no llegue la cultura informática y sea expandida para la superación y desarrollo del pueblo.

La UCI cuenta actualmente con alrededor de 10 000 estudiantes y 6000 trabajadores y al encontrarse en constante evolución maneja una gran cantidad de recursos tecnológicos, materiales de la construcción, escolares, etc. También desarrolla proyectos de importancia tanto nacional como internacional y trabaja en determinadas áreas con información clasificada lo que obliga a extremar las medidas de seguridad con el personal ajeno que accede a la misma. A este personal se le denomina visita, y se clasifica en: visita diurna o visita temporal. La diurna es aquella en la que el visitante permanece en el centro solo durante el horario laboral y la temporal cuando pernocta. Existen dos tipos de visitas temporales, aquellas que solicitan los profesores internos para familiares y las solicitadas por los directivos para hospedaje en el hotelito. Este flujo de personal se ha realizado diariamente, desde los inicios de la UCI, mediante un proceso de solicitud y autorización que se desarrolla de forma manual y personal que supone dificultades importantes, en cuanto a los factores disponibilidad, tiempo y seguridad.

Al no tener en funcionamiento un sistema que permita gestionar el flujo de solicitud y autorización de visitas a la UCI eficazmente y que con esto aumente el control de la seguridad física y de los bienes del centro, se plantea la siguiente **situación problemática**: las solicitudes y autorizaciones de visitas acarrear procesos que involucran a directivos del centro, solicitantes y trabajadores del Departamento de Seguridad y Protección. Todo este proceso requiere que el solicitante contacte físicamente a cada una de las personas involucradas, labor ésta que

consume bastante tiempo y genera, en ocasiones, no pocas incomodidades. Actualmente el acceso de visitantes no se controla de forma adecuada lo que provoca importantes lagunas en la seguridad del centro.

Este sistema tradicional de control de acceso basado en un registro manual de datos hace vulnerable la seguridad, los datos pueden resultar poco seguros y no confiables, además de requerir archivar información escrita a mano con poco o nulo aprovechamiento ya que esta es desechada cada cierto período de tiempo lo que hace difícil, y en ocasiones imposible, obtener reportes.

Todo lo antes expuesto demuestra la necesidad de crear un sistema que automatice los procesos de gestión de visita y la obtención de reportes.

El sistema eliminará los tediosos procesos asociados a la solicitud y autorización de visitas, haciéndolos más flexibles, ágiles y viables; aumentará la seguridad en el centro pues se podrá tener un control más exhaustivo del personal que la visita y permitirá obtener reportes con la información de todas aquellas personas que han visitado el centro, que lo visitarán o que se encuentran en él. Dicho sistema podrá ser accedido por los usuarios a través de la red desde cualquier punto de la universidad.

Se propone basar la investigación en el siguiente **problema científico**:

¿Cómo desarrollar un sistema que permita gestionar las solicitudes y autorizaciones de visitas a la UCI de una forma más fiable y organizada?

Así el **objeto de estudio** de este trabajo son los procesos de gestión de solicitud de visitantes.

El **campo de acción** de este trabajo se restringe a las solicitudes de visitas de personal ajeno a la universidad.

El **objetivo general** que persigue este trabajo es diseñar e implementar un sistema que controle el flujo de información y los procesos relacionados con la reservación y autorización del acceso de las visitas a la UCI así como brindar distintos servicios de reportes.

Como **objetivos específicos**:

- Fundamentar los temas relacionados con Sistema de Gestión de Visitas en la UCI, así como las tendencias en el orden internacional.
- Analizar las herramientas, lenguajes y metodologías de desarrollo a utilizar en el desarrollo del Módulo de Visitas.
- Documentar los requisitos necesarios para el cliente del sistema a desarrollar.
- Realizar el diseño y la implementación del Módulo de Visitas.

Para cumplir con los objetivos trazados se realizarán las siguientes **tareas**:

- ✓ Análisis de la bibliografía existente relacionada con el control de acceso de visitantes a instituciones.
- ✓ Estudio y caracterización de las herramientas, lenguajes y metodología de desarrollo a ser usadas en la solución.
- ✓ Realización de un estudio del entorno de trabajo del cliente.
- ✓ Entrevistas con el personal calificado de la oficina de Seguridad y Protección para determinar lo que el Módulo de Visitas requiere.
- ✓ Desarrollo del diseño adecuado para el sistema para su posterior implementación.

De lo antes expuesto se plantea la siguiente **hipótesis**: con la implementación del Módulo de Visitas para el Sistema de Control de Acceso, se podrá optimizar el proceso de solicitud y autorización de visitas en la UCI.

Variable independiente: Módulo de Visitas para el Sistema de Control de Acceso.

Variable dependiente: Gestión de visitas al centro

Los **métodos científicos** utilizados en la investigación fueron:

Métodos empíricos

- Método de observación: Se percibió directamente, sin intermediarios que deformaran la percepción, los hechos de la realidad objetiva, con lo cual se eliminaron las deformaciones de otros métodos indirectos.
- Método de entrevista: Se entrevistaron personas conocedoras del negocio y a través de ellas se alcanzó una visión más concreta del mismo.

También se pusieron en práctica los métodos teóricos siguientes:

- Analítico – sintético: Se analizaron las teorías, documentos, etc.; permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- Análisis histórico – lógico: Se constató teóricamente cómo ha evolucionado la gestión de visitas en un período de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo.
- Modelación: Se crearán modelos con vistas a investigar la realidad .Los modelos son mucho más fáciles de estudiar.

Como **posible resultado**: se obtendrá un sistema seguro y confiable que haga más sencilla la gestión de solicitud y autorización de visitas que realizan los trabajadores.

Este trabajo se ha estructurado de la siguiente manera:

Capítulo 1

Describe los principales problemas que presenta el actual proceso de reservación de visitas en la Universidad de las Ciencias Informáticas incluyendo un estado del arte de los sistemas que controlan estos procesos, de las tendencias, técnicas y tecnologías utilizadas en el mundo de hoy para desarrollarlos, así como la fundamentación de las metodologías y herramientas a utilizar.

Capítulo 2

Se incluye el objeto de estudio, la propuesta de sistema, el modelo de negocio, sus actores y trabajadores, además del diagrama de casos de uso del negocio y los diagramas de actividades, se identifican también los requerimientos y los casos de usos del sistema a automatizar.

Capítulo 3

Abarca todo lo referido al análisis y diseño del sistema a través del modelo de análisis como solución de la propuesta. Se describen los patrones de diseño utilizados para la confección de los diagramas de clase e interacción. Así como la arquitectura a utilizar en el desarrollo del sistema.

Capítulo 4

Se lleva a cabo la implementación del sistema y para ello se realiza la descripción y realización del diagrama de despliegue y el diagrama de componentes. Finalmente, concluye el capítulo con la realización de las pruebas correspondientes a los casos de uso significativos.

CAPITULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se llevará a cabo un estudio de la situación actual de los sistemas de gestión de visitas a nivel internacional, nacional y en la universidad. Se expone también la selección de las herramientas de desarrollo que permitirán la eficiente implementación del Módulo de Visitas así como la metodología que guiará el proceso. Se expondrá basada en la investigación la propuesta de solución a la situación problemática existente.

1.2 Necesidad del Módulo de Visitas para el sistema Control de Acceso

El control de acceso de visitantes en la universidad es responsabilidad del Departamento de Seguridad y Protección que tiene como objetivo principal proteger la integridad física de las personas, el equipamiento docente, tecnológico y para los servicios, las edificaciones con toda su infraestructura, los recursos y medios materiales para el desarrollo futuro de la universidad así como el patrimonio cultural. Diariamente se lleva a cabo un gran flujo de visitantes al centro con un promedio elevado de visitas diarias por lo que es muy difícil realizar una gestión rápida y segura de cada una. Sin un sistema que permita tener el control de forma exhaustiva de cada persona que se encuentra en la universidad y al mismo tiempo de quien visita la misma no se podría contar con un mecanismo de seguridad eficiente. En cualquier situación que pueda presentarse en la universidad es de vital importancia tener conocimiento de la cantidad real de personas que se encuentran en la misma. Todo esto sin contar la facilidad, en cuanto a tiempo y disponibilidad, que brindaría a los solicitantes y directivos facultados para autorizar las solicitudes, el poder realizar las mismas desde cualquier zona de la universidad.

1.3 Sistemas de Reservación de Visitas

Para obtener referencias que orienten la implementación del Módulo de Visitas para el Sistema de Control de Acceso se hace un estudio de sistemas vinculados al campo de acción. Actualmente los más populares son los de gestión de visitas online gracias a su facilidad de uso y seguridad. Existe una alta tendencia del uso de estos por empresas ya que constituyen soluciones que mejoran su productividad. Los sistemas de reservación de visitas en el ámbito internacional.

➤ **Sistema de gestión de visita**

Avances Tecnológicos es una empresa proveedora de soluciones informáticas, dedicada a la investigación de tecnologías emergentes y su implantación productiva en las empresas. Uno de sus productos es el Sistema de gestión de visitas, con el cual se le facilita a los empleados de una empresa la gestión de autorización de ingreso de sus visitantes, controlando además la entrada, tránsito y salida de estos últimos. Según la jerarquía organizacional del sistema, se produce el flujo automático para la aprobación de la solicitud, el nivel de jerarquía puede variar en dependencia de si se trata de un Visitante con auto, Visitantes sin auto, Visitas en fines de semana ó feriados, etc. El sistema también envía recordatorios en caso de que la persona que autoriza no haya tomado acción en un tiempo predefinido. La solicitud que haya sido aprobada se registra en la agenda del visitado y opcionalmente en la de otro empleado que haya tenido interacción con el visitante. El sistema muestra a los responsables de controlar el ingreso, las visitas autorizadas para ese día. Captura la imagen del visitante cuando accede por primera vez y esta esa usada como verificación adicional de la identidad. Notifica de los visitantes que están dentro de la empresa que aun no se han retirado. Brinda una serie de reportes organizados por diversos criterios. Corre en las plataformas: Windows (Intel), AIX (RS/6000), OS/400 (AS/400), OS/390 (S/390), OS/2, Red Hat Linux, Solaris. [1]

➤ **QAcces: Control de Visitas y Citas Previas**

QAccess es un sistema de Control de Visitas y Citas Previas para el seguimiento de visitantes en edificios públicos, de oficinas, fábricas y hoteles. Es un sistema fácil de instalar, configurar y utilizar. Identifica muy rápida y automáticamente a los visitantes mediante lectura directa de su DNI, pasaporte (u otro documento de identificación conforme a ICAO 9303), en segundos y sin errores de transcripción, simplemente deslizando el documento por la ranura de un lector OCR, y permite asignarles una tarjeta o identificación, agilizando su registro.

Puede asociarse con cualquier sistema existente de tarjetas de control. Opcionalmente archiva una imagen actual del visitante y/o genera una etiqueta autoadhesiva o autorización personalizada con el logotipo de la empresa. Proporciona información de las horas de entrada y salida de cada visitante así como de la persona o departamento visitado y número de la tarjeta de control asignada.

Suministra instantáneamente información acerca del número de visitantes que hay en cada momento en el (los) edificio(s), por zonas y/o plantas.

Controla el tiempo de desplazamiento de los visitantes desde el punto de acceso al punto visitado, lanzando mensajes de aviso cuando el tiempo estimado de desplazamiento, en cualquiera de los dos sentidos, ha sido superado.

Identifica a las personas que no han abandonado el edificio al final de la jornada.

Facilita información estadística de los horarios de afluencia de visitantes y de los tiempos medios de visita. En el caso de los hoteles facilita los listados diarios de altas de residentes. En los actos públicos permite la identificación de los asistentes de forma fácil y rápida. [2]

➤ **API-VISITAS**

El Sistema API-VISITAS permite realizar de manera segura, práctica y eficiente el control de todas aquellas personas ajenas a un centro, que ingresan y egresan del mismo. Su interfaz gráfica, sumamente amigable, agiliza la toma de datos y la captura de imágenes, disminuyendo al mínimo los tiempos de acceso. Toda esta información estará disponible en informes y estadísticas para un control y análisis con detalle de los lugares, fecha, horarios, tiempos de permanencia, persona visitada, periodicidad de las visitas, etc. [3]

➤ **OCEAN: Control de Accesos y visitas**

Ocean se encarga de la planificación, seguimiento y control de los accesos en un edificio o conjunto de edificios, sin importar la complejidad ni la distribución de los puntos a controlar. El Ocean permite registrar las visitas tanto en el caso en que se utilicen credenciales para limitar su acceso a ciertas partes de la empresa o planta, como en la variante en la que su tránsito es libre, sin necesidad de asignación de credenciales de acceso. El Ocean mantiene una base de datos permanente de las personas que fueron ingresadas en este módulo, permitiendo la búsqueda de sus datos mediante su apellido y nombre.

No se dedica de forma íntegra al control de visitantes sino que este es un módulo dentro del sistema. El sistema Ocean (C) Sage SRL 2001 se encuentra registrado como propiedad intelectual y no es libre. [4]

➤ **LobbyWorks**

El sistema de gestión de visitas LobbyWorks™ facilita la gestión de inscripción, identificación y gestiona el tráfico de visitantes, así como su seguimiento. La inscripción del visitante es rápida y puede efectuarse en la garita de vigilancia. Es un producto adecuado tanto para instalaciones

sencillas y autónomas utilizando la versión Professional como para instalaciones con varias localizaciones con la versión Standard. En general presenta las siguientes características:

- Inscripción previa de los visitantes esperados mediante interfaz web
 - Captura de la tarjeta de visita o DNI de los visitantes que llegan
 - Identificación automática de visitas ya reconocidas, visitas esperadas y visitas que figuran en una lista negra de vigilancia
 - Captura de la fotografía del visitante y/o la firma
 - Seguimiento de las tarjetas de acceso de visitante asignadas
 - Comprobación de los visitantes que requieran una aprobación especial
 - Impresión profesional en color de pases de visitante con fecha de caducidad, área de visita, anfitrión de la visita y objetivo de ésta
 - Notificación automática al empleado anfitrión de la llegada del visitante
 - Integración con sistemas de control de acceso para gestionar las acreditaciones para anfitriones y suministro de tarjetas de acceso a los visitantes
 - Generación rápida de informes detallados de tráfico de visitantes por empleado, departamento o edificio. [5]
- **Consortio Zona Franca de Vigo**

El Consorcio de la Zona Franca de Vigo es un territorio donde se aplica un régimen aduanero especial, con excepción de derechos arancelarios para las mercancías extranjeras y otras ventajas fiscales. Este consorcio brinda la posibilidad de visitar sus instalaciones para ello es necesario enviar los datos requeridos a través de su página web. En los formularios para la reservación es imprescindible ingresar de forma breve los motivos de la visita y un teléfono o correo electrónico para futuro contacto. [6]

1.3.1 Los sistemas de reservación de visitas en el ámbito nacional

En el ámbito nacional, los sistemas de gestión de visitantes se utilizan fundamentalmente en la esfera del turismo para las reservaciones hoteleras y viajes a través del país. Este sitio permite el acceso a los más exclusivos escenarios naturales de Cuba. Permite al usuario ver y seleccionar el programa adecuado a sus exigencias, entre los que se encuentra Hoteles, Pesca, Todo incluido, Náutica, Multidestinos, Naturaleza, Camper, Buceo y Caza así como la reserva de autos. [7]

- www.caribscuba.com
- www.dtcuba.com
- www.hotelescubanacan.com

1.3.2 Los sistemas de reservación de visitas en la UCI

Actualmente en la UCI no existe ningún sistema automatizado puesto en funcionamiento, que procese esta información y proporcione un riguroso control del personal ajeno al centro que accede a él, aunque en el año 2005 se desarrolló un trabajo de diploma sobre el tema del control de visitas a la Universidad conjuntamente con el de trabajadores cuyo título fue “Sistema de Control de Acceso”. Dicho trabajo fue realizado cuando la universidad contaba con un número menor de estudiantes y el control de las visitas y el acceso de los trabajadores no se realizaba de forma tan rigurosa o detallada como se requiere en estos momentos en que la Universidad cuenta con alrededor de 10000 estudiantes y 6611 trabajadores y que la cantidad de recursos materiales ha ido en aumento con la construcción de nuevos docentes y la ampliación de la beca. La propuesta de solución fue implementada con software propietario, lo que no se adapta a las actuales políticas de la universidad de migrar al software libre por sus grandes ventajas y posibilidades. En esta propuesta no se tuvieron en cuenta los temas de relacionados con la residencia y, al existir cambios en los requisitos poco después de su puesta en funcionamiento, el software dejó de utilizarse.

También en el año 2008 en otra tesis de grado titulada “Análisis y Diseño del sistema de Control de Acceso de la Universidad de las Ciencias Informáticas”, se analizó el control de las entradas y salidas de visitantes en la UCI, esta vez el sistema solo permitía la búsqueda del visitante en caso de que estuviera autorizado previamente utilizando como criterio de búsqueda su CI y registraba el acceso del mismo, así como de cualquier vehículo o medio que el visitante trajera. Esta es una tesis de Análisis y Diseño que aunque no abarca todos los procesos que se quiere con el presente trabajo si puede servir de referencia.

En la universidad funcionan sistemas de gestión que aunque no se adaptan al campo de acción, restringido a las solicitudes de visitas de personal ajeno a la universidad, conjugan elementos importantes de gestión que demuestran el uso de la infraestructura tecnológica para informatizar procesos de importancia para la Universidad. Estos sistemas están fundamentalmente dedicados a la reservación de transporte:

- pase.uci.cu

Es el encargado de la reservación del pase que se lleva a cabo por parte de los estudiantes los fines de semana. Brinda la posibilidad de escoger la ruta deseada.

- transportacion.uci.cu

Gestiona los servicios de transportación de profesores para los viajes semestrales, así como el servicio de pase masivo al finalizar el curso y en fin de año.

1.3.3 Motivos de su no selección

Como puede observarse existe variedad de sistemas que llevan a cabo la gestión de visitantes, desarrollándola de disímiles maneras y aportando diferentes servicios según la empresa que requiera la aplicación. Todos estos sistemas poseen muchas facilidades de uso, son sencillos de instalar y poseen el control casi total del personal trabajador o visita en una empresa pero, estos sistemas no son apropiados, a pesar de lo muy efectivos y las muchas ventajas que presentan, debido a que, para utilizarlos se debe comprar su licencia, y esto no se adapta a las políticas específicas de la UCI en cuanto a temas de soberanía tecnológica. En su mayoría no son multiplataforma. Para su implantación es preciso entrenamiento técnico, así como un conjunto de elementos de software, hardware y accesorios que implicarían un elevado costo tanto de adquisición como de mantenimiento. Los servicios específicos que se requieren con la implementación de Módulo de visitas para el Sistema de Control de Acceso no se abarcan en su totalidad en ninguno de los sistemas estudiados.

1.4 La aplicación

El despliegue y mantenimiento del software que no utiliza la web y que es basado en una arquitectura cliente-servidor resulta una tarea bastante compleja y engorrosa; para su instalación es necesario ir a cada usuario que utilizará dicho software e instalar éste en cada máquina cliente, esto trae consigo numerosos inconvenientes, por ejemplo, una mejora en el servidor, como parte de la aplicación, requiere normalmente una mejora de los clientes instalados en cada ordenador personal, añadiendo un coste de soporte técnico y disminuyendo la productividad; sin contar además que actualmente las máquinas que ejecutan este tipo de aplicaciones deben disponer de programas que no consumen pocos recursos, como por ejemplo la JVM y el .Net Framework.

Las aplicaciones web eliminan todo este tipo de inconvenientes, ellas no requieren de clientes con las prestaciones señaladas anteriormente, hacen uso sólo de clientes ligeros, donde no es necesario instalar la aplicación, la cual se ejecutará en un servidor (servidor web o servidor de aplicaciones) mejorando en gran medida el soporte y la actualización del software, realizándose de forma única y centralizada. El cliente tan sólo con un navegador web que interprete el código HTML o XHTML generado por el servidor web y el código Java Script que utilice la aplicación del lado del cliente obtendrá a través de internet o una intranet todas las funcionalidades que brinda dicho software.

La finalidad de la aplicación propuesta es agrupar una serie de elementos que permitan al usuario acceder de forma sencilla y segura a las funcionalidades que brinda el sistema a través de la web, así como gestionar de forma interactiva y ágil las solicitudes de visitas y tener acceso a una serie de reportes que deberán almacenarse en una base de datos; siendo estas las razones principales para la elección de una aplicación web para la realización del Módulo de Visitas para el sistema de Control de Acceso.

1.5 Caracterización de las herramientas

Para el desarrollo de aplicaciones web existen una gran cantidad de herramientas de las que se estarán analizando algunas para elegir las más acordes para la implementación del Módulo de Visitas, se requiere que todo el software utilizado sea libre por las amplias ventajas que esto proporciona. Se puede ver por ejemplo:

Software Libre

Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Entre las amplias posibilidades que brinda se encuentran:

Escrutinio público: Al ser muchas las personas que tienen acceso al código fuente, este se somete a un proceso de corrección de errores muy dinámico, no es necesario que el proveedor del software saque una nueva versión.

Independencia del proveedor: Al disponer del código fuente, cualquier persona puede continuar ofreciendo soporte, desarrollo, u otro tipo de servicios para el software lo que garantiza su continuidad.

Manejo de la lengua (traducción): Cualquier persona capacitada puede traducir y adaptar un software libre a cualquier lengua. Corrección ortográfica y gramatical.

Mayor seguridad y privacidad: Los sistemas de almacenamiento y recuperación de la información son públicos. Cualquier persona puede ver y entender cómo se almacenan los datos en un determinado formato o sistema. Existe una mayor dificultad para introducir código malicioso como ser: espía (p/ej. capturador de teclas), de control remoto (p/ej. troyano), de entrada al sistema (p/ej. puerta trasera), etc.

Ahorro en costos: en cuanto a este tópico se distinguen cuatro grandes costos: de adquisición, de implantación (este a su vez se compone de costos de migración y de instalación), de soporte o mantenimiento, y de interoperabilidad. [8]

1.5.1 Lenguaje de programación

PHP significa: Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por El Grupo PHP (The PHP Group). PHP es el lenguaje más utilizado en la web, es un lenguaje interpretado, de propósito general, ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Es el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web, puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. La versión php5.2.8, una de las más recientes versiones del 08 de diciembre de 2008, es la que se utilizará en el desarrollo del sistema. Esta versión ha mejorado muchas carencias de los mecanismos de POO de versiones anteriores y ahora tiene una mejor capacidad para expandir su potencial utilizando la enorme cantidad de módulos. Presenta una biblioteca nativa de funciones muy amplia y un sólido manejo de excepciones, además de que es respaldado por una fuerte comunidad. Está publicado bajo una licencia de software libre. (Licencia PHP License 3.01).

1.5.2 Plataforma de desarrollo

Un framework es una plataforma de trabajo que facilita la programación de aplicaciones, permitiendo a los diseñadores y programadores centrarse en los requerimientos específicos de la aplicación con ello el desarrollo web se hace más intuitivo y las aplicaciones resultantes son más robustas. Se han tomado en cuenta como posibles plataformas a utilizar:

Zend Framework es utilizado en el desarrollo de aplicaciones web y servicios web con PHP 5. Brinda soluciones para construir sitios web modernos, robustos y seguros. Es de código abierto, utiliza licencia BSD. Es extremadamente simple consultar su base de datos, sin tener que escribir ninguna consulta SQL. Utiliza arquitectura MVC. Este framework cuenta con robustas clases para autenticación y filtrado de entrada, así como clientes para servidores web. En lo que a documentación se refiere, no posee demasiadas fuentes, siendo ésta una desventaja importante.

Symfony 1.2 pertenece al conjunto de framework basados en el MVC, está desarrollado en PHP 5 y diseñado para optimizar el desarrollo de las aplicaciones web. Entre sus características, se pueden enunciar que es fácil de instalar y configurar en la mayoría de las plataformas y es independiente del sistema gestor de bases de datos. Posee un potente mecanismo de cache y de ruteo, además de poseer una gran integración con Ajax. Es muy sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones, aquí se incluye el framework para formularios que está presente desde la versión 1.1 y que es una de las creaciones más robusta y creativas de Symfony.

¿Por qué se selecciona Symfony?

En cuanto a documentación y aprendizaje Symfony dispone de guías y manuales en abundancia, y además cuenta con numerosos foros en varios idiomas. En el caso de Zend la comunidad es algo escasa, por tanto su documentación también. Y otro tanto ocurre con los plugins, en Symfony es posible aumentar sus funcionalidades a base de plugins, mientras que Zend carece de esta característica. El trabajo con base de datos en Zend se limita a usar ActiveRecord, pero en Symfony tienes la posibilidad de usar el motor de base de datos que quieras, incluyendo el propio Zend_Db, algo que aporta una enorme flexibilidad ya que brinda la posibilidad de elegir en cada proyecto cual es la mejor opción. Se ha seleccionado Symfony en su versión 1.2 como framework de desarrollo para el sistema.

1.5.3 Sistema Gestor de Base de Datos

Hoy en día, prácticamente todo software de gestión necesita hacer persistente y tener acceso a su, en muchas ocasiones, amplio volumen de información de una forma óptima y segura. Los

sistemas gestores de bases de datos son los programas encargados de realizar ésta y muchas otras funciones más; por consiguiente el presente sistema también debe hacer uso de ellos. Se han tomado en cuenta como posibles sistemas gestores de base de datos a utilizar:

MySQL es, probablemente, el gestor de bases de datos más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debido, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación (C++, Java, PHP, etc.), además de su fácil instalación y configuración. Además posee gran fortaleza en la gestión de usuarios y contraseñas manteniendo un muy buen nivel de seguridad en los datos. MySQL fue diseñada para ser un gestor de bases de datos con una gran rapidez de respuesta. Fue adquirido por Sun Microsystems, ahora está patrocinado por una empresa propietaria, que posee el copyright de la mayor parte del código.

PostgreSql es un potente sistema gestor de base de datos relacional orientado a objetos de software libre, publicado bajo la licencia BSD, desarrollado por PostgreSQL Global Development Group traducido al español como Grupo Global de Desarrollo de PostgreSQL. PostgreSQL tiene un activo desarrollo y una probada arquitectura que se ha ganado una sólida reputación de fiabilidad, es una excelente solución para la administración de la base de datos. Funciona en todos los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. La nueva versión 8.3 incluye una cantidad récord de características nuevas y mejoradas respecto a versiones anteriores. La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes. A su vez, posee una gran escalabilidad y es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta, acoplándose, fundamentalmente esta característica a las necesidades del sistema a desarrollar.

¿Por qué se selecciona PostgreSql?

Se escogió PostgreSql porque es realmente la mejor alternativa libre, por no decir la única que se tiene hoy en día.

1.5.4 Entorno integrado de desarrollo

Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE cuenta con un editor de código, un compilador/intérprete y un depurador.

NetBeans 6.5 es una herramienta que facilita el desarrollo de aplicaciones. A pesar de estar escrito en Java ofrece soporte para otros lenguajes entre los que se encuentra PHP. Como una de las principales características de su selección es que es de código abierto y gratuito, tanto para uso comercial como personal, bajo licencias como GPL. NetBeans es multiplataforma y está disponible para varios sistemas operativos entre los que se encuentra Windows.

La nueva versión 6.5, la cual incluye numerosas mejoras que benefician el buen funcionamiento de la aplicación resultante, brinda mejoras en la gestión de bases de datos, la compilación de los proyectos se realiza mucho más rápido además de brindar la posibilidad de buscar funcionalidades del IDE utilizando un buscador.

1.5.5 Servidor Web

Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP.

Servidor Web Apache 2.0 posee facilidad de configuración, robustez, estabilidad y transparencia. Apache está desarrollado por la Fundación de Software Apache (Apache Software Foundation), es una tecnología gratuita de código fuente abierto con licencia descendiente de las licencias BSD, por lo que se puede hacer uso de su código fuente. Lo que hace a este servidor web universal, es que corre en una multitud de sistemas operativos. Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a éste. Brinda soporte para varios lenguajes: PHP, JAVA, Perl y librerías ASP. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Un gran número de aplicaciones web están montadas en servidores Apache. El siguiente gráfico muestra los resultados de un estudio hecho por Netcraft, empresa dedicada a encuestas a nivel global y estudios sobre el tráfico en Internet:

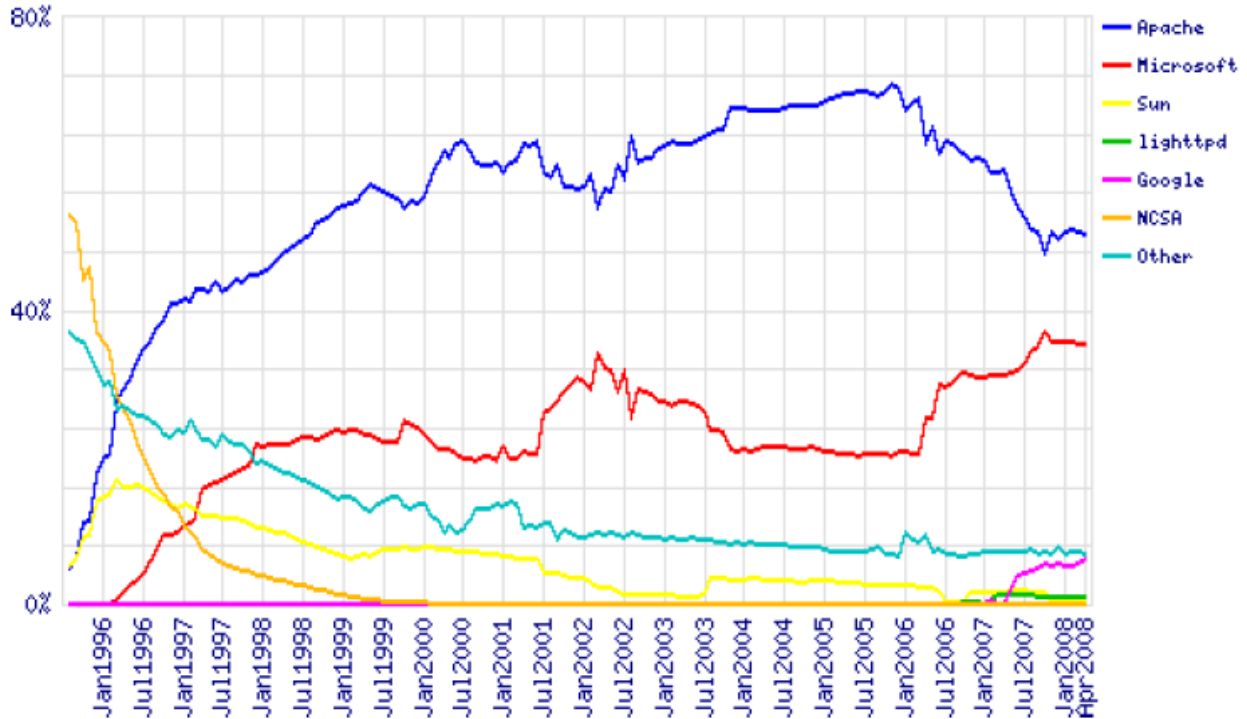


Figura 1 Análisis realizado por Netcraft de la explotación de Servidores Web

Microsoft IIS 6.0 (Internet Information Services IIS), es una serie de servicios para los ordenadores que funcionan con Windows. Se incluye en Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Este servicio convierte a un ordenador en un servidor de Internet o Intranet, es decir, que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web). El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, ya sean ASP y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

La versión 6.0 es la más empleada. Los principales avances de esta última versión vienen motivados sobre todo por la seguridad y el rendimiento, aunque todavía sufre de algunos agujeros de seguridad. Es capaz de impedir que aquellos usuarios con direcciones IP conocidas obtengan acceso no autorizado al servidor. La administración de la seguridad del servidor IIS es una tarea fácilmente ejecutable a base de asistentes para la seguridad.

¿Por qué se selecciona Apache 2.0?

Se debe tener en cuenta que Apache ha sido por más de 13 años consecutivos el servidor web de mayor utilización alrededor del mundo, por lo que existe una vasta experiencia y documentación y ha demostrado con creces desde su nacimiento su estabilidad, solidez y rendimiento superior. Ser más seguro es una cuestión de complejidad lo que provoca que los servidores web que usan Software Libre (GNU/Linux+Apache) son más seguros que los que corren Software Privativo (Windows+IIS) según un estudio realizado por Google donde se demostró que en menos porcentaje (IIS 23%) se ha encontrado el mismo número de malware que en un mayor porcentaje (Apache 49%). La principal desventaja de IIS es que no es software libre mientras que Apache es un proyecto libre, y por tanto, gratuito, por lo que se selecciona Apache como servidor web a utilizar.

1.5.6 Lenguaje Unificado de Modelado

UML: Lenguaje Unificado de Modelado (Unified Modeling Language), es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. No es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso; es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. [9]

UML puede usarse para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. Es un lenguaje de propósito general para el modelado orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje común para todos los desarrolladores, por lo que cuenta con una documentación también común, sin importar el software con que se desarrolle tampoco especifica en sí mismo qué metodología o proceso usar. Es sencillo de aprender y de utilizar. Permite especificar todas las decisiones de análisis, diseño e implementación, lo que permite construir modelos precisos y completos.

1.5.7 Herramienta CASE

Las herramientas CASE (Computer Aided Software Engineering, traducido al español como: Ingeniería de Software Asistida por Ordenador) son aplicaciones destinadas a crecer la productividad en el desarrollo del software. Asisten al desarrollador en todas las etapas del ciclo de vida del proyecto fundamentalmente en la planificación

Visual Paradigm UML 6.4 Enterprise Edition es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a la rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. Esta herramienta también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Permite la generación de bases de datos. Permite ingeniería inversa de bases de datos, desde sistemas gestores de bases de datos existentes a diagramas de entidad-relación. Es un software de código abierto.

Rational Rose 7.0 Enterprise es el producto más completo de la familia Rational Rose, por lo que incluye soporte para UML. Proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad. Esta versión incluye los objetos más recientes de UML además de diagramas de casos de uso, diagramas de clase, diagramas de componentes, integración con Microsoft Visio, además permite generar reportes y documentación en HTML/PDF. IBM Rational Rose Enterprise se puede adquirir con licencias de usuario autorizado (usuario único) o con licencias flotantes (usuarios simultáneos), ya que es de software propietario.

¿Por qué se selecciona Visual Paradigm UML 6.4 Enterprise Edition?

Ambas herramientas proporcionan ventajas que son de utilidad en la confección del sistema ya que brindan el apoyo necesario en cada una de las fases del ciclo de desarrollo. La principal por la cual no se utiliza Rational Rose es porque no pertenece al grupo de software libre, y es necesario pagar por su uso, en cambio para Visual Paradigm la universidad posee la licencia.

1.5.8 Fundamentación de la metodología

1.5.8.1 Metodologías ágiles

Los procesos ágiles de desarrollo de software, intentan evitar los caminos de las metodologías tradicionales y enfocarse en la gente y los resultados. Estas metodologías plantean principalmente según el Manifiesto Ágil que los individuos e iteraciones son más importantes que los procesos y herramientas, un software que funcione es más importante que una documentación exhaustiva, la colaboración con el cliente es más importante que la negociación

de contratos y la respuesta ante el cambio es más importante que el seguimiento de un plan. Entre las metodologías ágiles se encuentran:

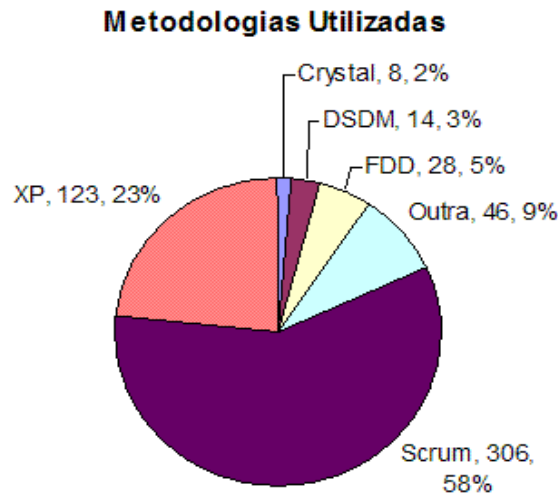


Figura 2 Porcentaje de utilización de las diferentes metodologías.

Scrum incentiva el trabajo de equipos pequeños y auto dirigidos y se basa en el control sistemático de las soluciones. El ciclo de vida de esta metodología es incremental e iterativo y en cada iteración se deberá entregar al cliente una solución implementada. Se enfoca en desarrollar las funcionalidades que realmente van a ser de utilidad para el cliente, asegurando obtener un producto en menos tiempo al evitar el trabajo innecesario. Está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. Sus elementos fundamentales son los Sprint que están formados por la serie de iteraciones que se llevaran a cabo durante la vida del proyecto y el Product Backlog que es la lista de funcionalidades que necesita el cliente y sirve de herramienta de referencia para el equipo.

Los roles en Scrum son pocos: el *product owner* es el responsable del proyecto; el *Scrum master* cuyo rol le pertenece al Líder del equipo o Gerente del proyecto; el *Customer* es el cliente y el *Scrum team* que lo constituye el equipo de proyecto, este es auto-organizado y usualmente está compuesto entre 4 y 8 miembros que trabajan de forma cohesionada.

Programación Extrema (XP) metodología basada en una serie de principios, valores y prácticas que la guían. Surge como respuesta a los problemas derivados del cambio en los requisitos, es una metodología a utilizar en proyectos de riesgo. Fue creada para evitar los problemas más comunes en las demás metodologías, referidos a que los proyectos fracasan porque exceden el

tiempo previsto para su concepción o, al finalizar, el mismo no cumple con las verdaderas necesidades del cliente ya que estas han estado variando durante el desarrollo del proyecto. XP se basa fundamentalmente en las pruebas unitarias, que se le realizan a los principales procesos para obtener posibles errores. También en la re fabricación, que se basa en la reutilización del código; y en la programación en pares, que consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Los valores fundamentales de esta metodología son: comunicación, simplicidad, realimentación y coraje; y sus principios: realimentación veloz, modificaciones incrementales, trabajo de calidad, asunción de simplicidad.

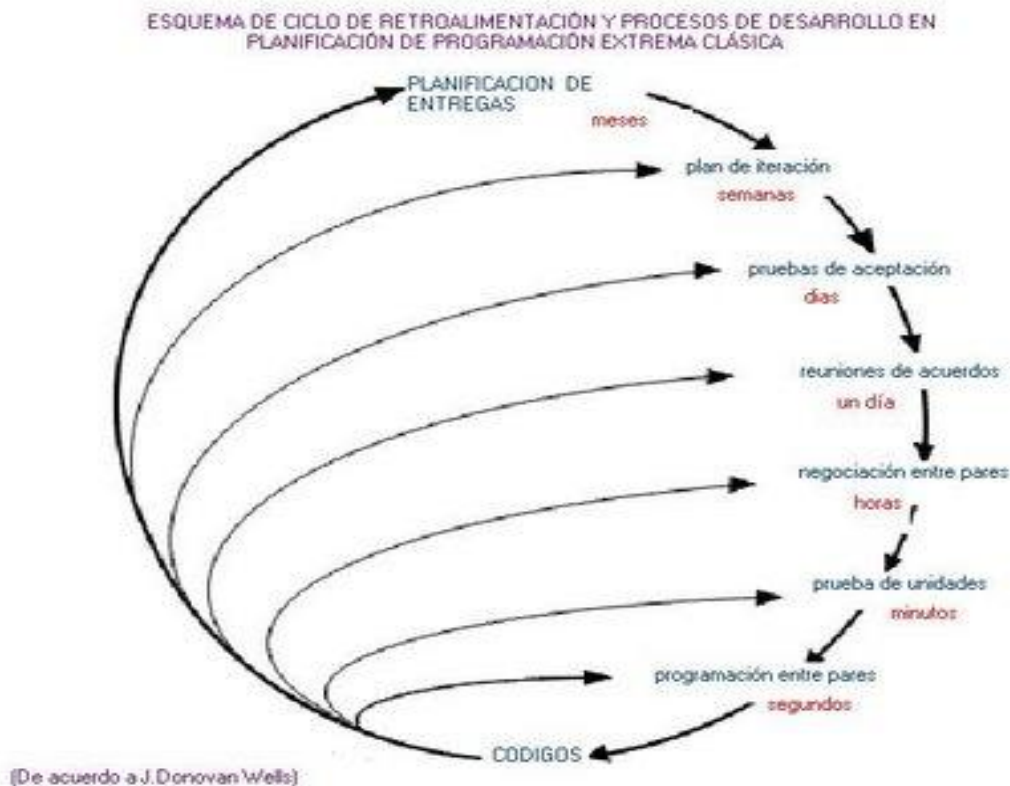


Figura 3 Ciclo de retroalimentación y procesos de desarrollo XP.

1.5.8.2 Metodología pesada

Proceso Unificado de Desarrollo

RUP es uno de los procesos más generales existentes en la actualidad. El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. Un proyecto realizado utilizando RUP se divide en cuatro fases: Inicio, Elaboración, Construcción y Transición. (Jacobson, y otros)

El ciclo de vida se desarrolla dividido en dos disciplinas: disciplina de desarrollo y disciplina de soporte. La primera se despliega en negocio, requerimientos, análisis y diseño, implementación y pruebas. La segunda configuración y administración del cambio, administrando el proyecto, ambiente y distribución.

RUP es una metodología para proyectos que cuenten con un amplio equipo de trabajo que estará dividido por roles bien definidos. Genera un extenso volumen de información bien detallada.

¿Por qué se selecciona RUP?

Anteriormente se expusieron algunas de las metodologías de desarrollo del software y por el estudio realizado se llega a la conclusión de que la que se utilizara para la aplicación propuesta será la metodología RUP ya que:

- Es una metodología exitosa de probada efectividad en muy variables proyectos puestos en práctica internacional y nacionalmente.
- Suficiente documentación que facilitará la mejor comprensión del sistema en caso de anexo a otros módulos del Sistema de Control de Acceso.
- Se necesita una metodología bien organizada, con una eficaz captura de requisitos que responda a los intereses del cliente.
- Esta metodología brinda una amplia y rica gama de artefactos que pueden ser usados según las necesidades del sistema y que guían detalladamente el desarrollo del software.
- Es la metodología que se imparte en el plan de estudio de la UCI y de la que existe una amplia y fiable documentación que facilita su estudio y puesta en práctica.

1.6 Propuesta de solución

Se propone el desarrollo de una aplicación web que automatice todos los procesos de solicitud de visitas que tienen lugar en la universidad y que en estos momentos se realizan de forma manual, acarreando una serie de incomodidades a las personas involucradas en estos procesos. Dicho sistema podrá ser accedido desde cualquier punto de la UCI por cualquier usuario autorizado a solicitar pases a visitantes.

La aplicación tendrá una base de datos propia que almacenará todos los registros asociados a su negocio, de esta manera se tendrá un riguroso control de la información que se genera en estos procesos. El mismo consumirá servicios web.

El sistema constará de un fuerte componente administrativo que permitirá su mantenimiento, su fiabilidad y su flexibilidad ante los cambios del negocio que puedan surgir.

A través de los reportes que generará la solución se podrán hacer estudios, controles y llevar estadísticas útiles tanto para el personal de Seguridad y Protección de la universidad como para los directivos.

Se pretende además que cada funcionalidad que brinde la aplicación pueda ser llevada a cabo sólo por el usuario que le está permitido esto, llegando a alcanzar un alto grado de seguridad.

1.7 Conclusiones

En este capítulo se demostró la necesidad de implementar una aplicación que agilice y controle los procesos de gestión de solicitudes de visitas debido a que, los sistemas estudiados relacionados con el campo de acción no abarcan la totalidad de funcionalidades requeridas. Se expuso la elección de implementar el Módulo de visitas como una aplicación web de interfaz amigable y posible acceso desde cualquier punto de la universidad. Se fundamentó el uso de software libre por sus ventajas y la utilización de PHP como lenguaje de programación, Symfony como framework, NetBeans como IDE, PostgreSQL como gestor de base de datos, como servidor web Apache y como herramienta CASE Visual Paradigm. La metodología escogida es RUP por las amplias posibilidades que brinda frente a las metodologías ágiles.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se detallará el entorno de trabajo, definido por el objeto de estudio, donde el sistema funcionará. Primeramente se realizará la modelación del negocio, para describir y definir los actores y trabajadores implicados y su participación en los procesos. Se valorarán los requisitos funcionales y no funcionales para definir luego los casos de uso del sistema. Se definirán los actores que van a interactuar con la aplicación. Se efectúa la estimación del esfuerzo y el análisis de costo y beneficios.

2.2 Reglas del Negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. [10]

Primeramente, se explicarán algunos conceptos asociados al modelo del negocio:

Visita diurna: Será toda aquella visita que se realice al docente o a cualquier área productiva de la universidad en el horario de 8:00am a 5:00pm, autorizadas por el Directivo del área correspondiente.

Pase: Documento impreso que recoge todos los datos del visitante.

Visita temporal: La visita temporal solo la puede solicitar un trabajador interno de la universidad. Solo puede solicitar autorización para madre, padre, hermana(o), tía(o), prima(o), esposa(o). Como requisito fundamental deben tener al menos un apellido en común. El tiempo máximo es de 72 horas y en caso de prórroga solo es autorizada por el Vicerrector de Residencia.

Hospedaje en el Hotelito: La solicitud de hospedaje la realizará el Vicerrector, Directivo o Decano responsable de la presencia del huésped en la UCI, en todos los casos se hace al menos con 5 días de antelación a la fecha inicial de hospedaje.

- Cuando la solicitud de reservación sea igual o mayor a los 90 días-hombre, serán aprobadas por el Rector.
- Las solicitudes de prórroga de huéspedes por más de 7 días, serán aprobadas por el Rector.

Solicitud: Permiso que se solicita para la entrada de personas ajenas a la universidad. Las solicitudes tienen tres estados: Pendiente, Aprobada y Denegada.

Visita: Una solicitud aprobada es una visita.

En el caso de los visitantes a los que les sea autorizada la entrada tendrán distintos estados según el proceso en el que se encuentren:

- Pendiente: el si el visitante aun no ha accedido a la universidad.
- En curso: el visitante se encuentra dentro del centro.
- Terminada: el visitante ya ha abandonado el centro.
- No presentado: la fecha de salida ha pasado y el visitante no se ha ingresado.
- Ilegal: la fecha de salida ha pasado y no se ha registrado la salida del visitante.

En caso de que la solicitud no sea aprobada el estado del visitante seria, denegado.

2.3 Modelo de negocio

El modelo del negocio se realiza con el propósito de intuir las características y actividades que se llevan a cabo en el contexto a automatizar. [10] En él, se modelan los clientes y trabajadores del negocio y su interacción con los procesos y objetos que lo componen. Está formado por el modelo de casos de uso del negocio y el modelo de objetos del negocio. A continuación se describe textual y gráficamente las etapas del modelo del negocio.

2.3.1 Definición de actores del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [10]

Actores del negocio	Descripción
Huésped	<ul style="list-style-type: none"> ➤ Especialistas nacionales o extranjeros, que no siendo trabajadores de la plantilla UCI o residentes de Ciudad de la Habana se encuentran en función de misiones o proyectos productivos. ➤ Cuadros, directivos, especialistas, y profesores de las Facultades Regionales en funciones de trabajo

	<p>en la UCI.</p> <ul style="list-style-type: none"> ➤ Cuadros, directivos o especialistas de otros organismos o empresas que por razones de trabajo requieran estancia en la UCI, solicitado por un Vicerrector y previamente autorizado por el Rector. ➤ Participantes en eventos que se desarrollen en la UCI, previamente aprobada por la Rectoría o en comisiones creadas al efecto. ➤ Otros casos autorizados por la Dirección del Proyecto o el Rector.
Solicitante	Es todo aquel trabajador que pertenezca a la universidad.
Trabajador Interno	Es aquel trabajador que reside en la universidad.

Tabla 2.1 Descripción de actores del negocio.

2.3.2 Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. [10]

Trabajadores del negocio	Descripción
Directivo	Trabajador de la UCI con cargos en la dirección de la universidad, responsable de determinada área que puede aprobar solicitudes de visita a su área y solicitar el hospedaje de un huésped en el hotelito de la UCI.
Director de Seguridad y Protección.	Máximo directivo encargado del área de Seguridad y Protección de la universidad.
Director de Residencia de Profesores.	Directivo responsable del área de residencia de profesores de la universidad.
Vice rector Residencia.	Máximo directivo del área de residencia de la universidad.
Rector.	Máximo directivo de la universidad.

Instructora.	Técnico C en atención a profesores.
Técnico S/P.	Responsables de controlar la entrada de personas, vehículos y medios a la universidad. Confecciona un pase de entrada para las visitas diurnas.
Especialista Superior de Residencia de Profesores.	Encargado de atender determinados elementos en el área de la residencia.
Especialista de Seguridad y Protección.	Encargado del control de todas las vistas que se realizan en la universidad.

Tabla 2.2 Descripción de trabajadores del negocio.

2.3.3 Diagrama de casos de uso del negocio

El diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. [10]

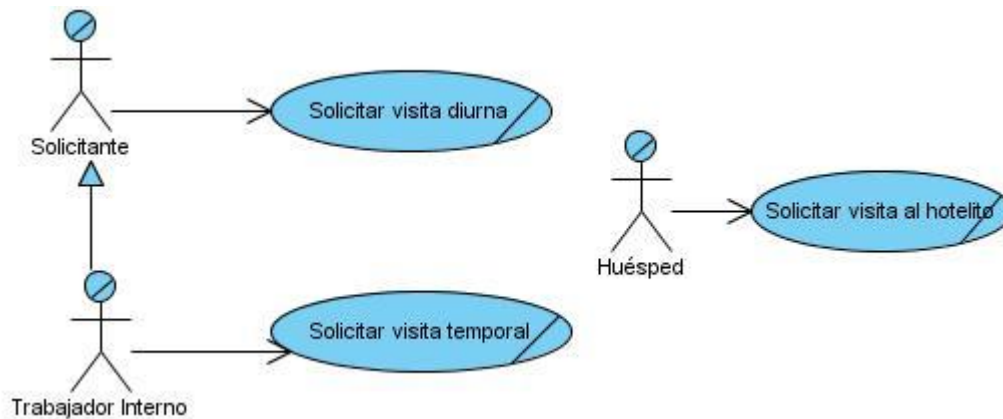


Figura 4 Diagrama de casos de uso del negocio.

2.4 Descripción textual de los casos de uso del negocio

La descripción textual de los casos de uso del negocio ayudara a comprender mejor la lógica del mismo.

Caso de Uso.	Solicitar visita diurna.
Actores del negocio.	Solicitante.
Trabajadores del negocio.	Directivo, Especialista de Seguridad y Protección, Director de Seguridad y Protección, Técnicos S/P

Propósito.	Realizar la solicitud de entrada de visitas diurnas a la universidad.
Resumen	
El caso de uso se inicia cuando el solicitante realiza la solicitud a su directivo y finaliza cuando la misma es aprobada o denegada y llega la información a la garita de acceso.	
Precondiciones.	El solicitante debe ser trabajador o estudiante de la universidad.
Flujo normal de eventos.	
Acción del actor.	Respuesta del negocio.
1- El solicitante se presenta en la oficina del directivo realizando la solicitud.	
	2-El Directivo valora la solicitud y determina que puede ser aprobada. 3-Solicita los datos del visitante.
4- Informa todos los datos de la visita.	
	5-Si la visita es imprevista informa los datos por vía telefónica al Técnico SP. 6- El Técnico SP verifica los datos informados con el carnet de identidad de la visita. 7- Si estos coinciden confecciona el pase de entrada y registra al visitante en el Registro de Visitantes.
8-El solicitante recibe información y se retira de la oficina del directivo.	
Pos condiciones	Es realizada la solicitud y la visita diurna es llevada a cabo.
Mejoras	La automatización de este caso de uso reducirá el tiempo de respuesta y permitirá mejorar la gestión del proceso. Un solicitante podrá acceder a sus solicitudes para verificar el estado de la misma (aprobada, denegada, pendiente). No habrá que recurrir personalmente ante el Directivo.
Prioridad	Alta
Flujos alternos de trabajo	

Acción del actor	Respuesta del negocio
	2.1 El Directivo analiza la solicitud, la deniega e informa al solicitante. (Ir al paso 8).
	5.1 En caso de que la visita no sea imprevista la añade al listado de visitas programadas. 5.2 Envía esta información al Director de Seguridad y Protección el cuál la reenvía al Especialista de Seguridad y Protección. 5.3 Este imprime el listado y lo hace llegar a manos del Técnico SP concluyendo así el caso de uso.
	6.1 En caso de no concordar le deniega la entrada a la universidad. (Ir al paso 6).

Tabla 2.3 Descripción textual del CU Solicitud de visita diurna.

Caso de Uso.	Solicitar visita temporal.
Actores del negocio.	Trabajador interno.
Trabajadores del negocio.	Especialista Superior de Residencia de Profesores, Director de Residencia de Profesores, Instructora.
Propósito.	Que se realice la gestión de solicitud de autorizo para acceder a la residencia de la universidad.
Resumen	
El caso de uso se inicia cuando el solicitante accede a la Oficina de Residencia de Profesores y realiza su solicitud de visita temporal, terminando el mismo cuando esta es aprobada o no.	
Precondiciones.	El solicitante debe ser interno.
Flujo normal de eventos.	
Acción del actor.	Respuesta del negocio.
1- El trabajador interno arriba a la Oficina de Residencia de Profesores e informa que desea realizar una solicitud de visita temporal, se identifica e informa los datos del visitante.	

	<p>2- Si la solicitud no es de prórroga la Instructora valora la solicitud del profesor interno que la solicita.</p> <p>3- Anota en el Registro de Solicitudes de Visitas los datos referentes a esa visita.</p> <p>4- Le confecciona el pase temporal de entrada.</p> <p>5- El pase temporal de entrada es firmado por el Director de Residencia de Profesores y por el Especialista Superior de Residencia.</p> <p>6- El pase es entregado a la Instructora la cual se encarga de entregarlo al solicitante.</p>
7- El solicitante se retira.	
Pos condiciones	Se realiza la solicitud de visita a la residencia.
Mejoras	La automatización del caso de uso permitirá agilizar la gestión de autorizaciones, brindando una mayor comodidad a los involucrados.
Prioridad	Alta
Flujos alternos de trabajo	
Acción del actor	Respuesta del negocio
	<p>2.1- Se detecta algún problema en la solicitud, ésta se deniega e informa al trabajador interno. (Ir al paso 7).</p> <p>2.2- Si la solicitud es de prórroga la instructora remite la solicitud al Director de Residencia.</p> <p>2.3 El Director de Residencia valora la solicitud y la aprueba(Ir al paso 3)</p> <p>2.3.1 El Director de Residencia no aprueba la prórroga e informa al solicitante (Ir al paso 7).</p>

Tabla 2.4 Descripción textual del CU Solicitar visita temporal.

Caso de Uso.	Solicitar visita al hotelito.
--------------	-------------------------------

Actores del negocio.	Solicitante.
Trabajadores del negocio.	Vicerrector de Residencia, Rector y Directivo
Propósito.	Que el huésped realice la solicitud de visita con hospedaje en el hotelito.
Resumen	
El caso de uso se inicia cuando el huésped solicita hospedaje en el hotelito. Esta solicitud es aprobada o denegada.	
Precondiciones.	Debe ser un huésped que cumpla con las condiciones especificadas por la Rectoría.
Flujo normal de eventos.	
Acción del actor.	Respuesta del negocio.
1- El solicitante pide hospedaje al Directivo.	
	2. El Directivo confecciona la Solicitud de hospedaje para profesores y especialistas invitados y la envía al Vicerrector de Residencia si el hospedaje no es por más de 90 días-hombre o prórroga de más de 7días. 3-El Vicerrector de Residencia analiza la solicitud e informa al Directivo de la autorización de la misma.
4- El solicitante recibe información.	
Pos condiciones	Se gestiona la solicitud de huéspedes al hotelito de la universidad.
Mejoras	La automatización de estos procesos permitirá que Directivos de la universidad realicen estos trámites de forma sencilla y eficaz, ahorrando tiempo y gestiones para la aprobación de su solicitud.
Prioridad	Alta
Flujos alternos de trabajo	
Acción del actor	Respuesta del negocio

	<p>2.1 -Si la solicitud es por más de 90 hombre-día o prórroga, se le envía directamente al Rector de la universidad.</p> <p>2.1.1 El Rector analiza la solicitud y determina que la visita no debe hospedarse en el hotelito e informa al Directivo el cuál informa al visitante. (Ir al paso 5).</p> <p>3.1-El Rector analiza la solicitud e informa al directivo de la autorización de la misma.</p>
	<p>2.2 El Vicerrector de Residencia analiza la solicitud y determina que el huésped no debe hospedarse en el hotelito e informa al Directivo el cuál informa al visitante. (Ir al paso 5).</p>

Tabla 2.5 Descripción textual del CU Solicitar visita al hotelito.

2.5 Diagramas de actividades

El diagrama de actividad es un grafo (grafo de actividades) que contiene estados en que puede hallarse una actividad. [10]

En estos diagramas se señalan las principales actividades a automatizar. Estos elementos son esenciales en el funcionamiento del negocio, por lo que fueron seleccionadas como funcionalidades básicas a desarrollar en la confección del sistema. En el anexo #1 se muestran los diagramas de actividades correspondientes a los casos de uso descritos.

2.6 Modelo de objetos del negocio

El modelo de objetos del negocio describe como colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio. [10]. Ver figura en anexo #2.

2.7 Relación de los requerimientos

Para lograr un sistema automatizado con eficiencia y calidad se debe tener en cuenta los requisitos funcionales y no funcionales. La captura de requisitos constituye un paso

fundamental para desarrollar cualquier sistema. A continuación se muestran las especificaciones encontradas durante el proceso.

2.7.1 Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Estos no alteran la funcionalidad del producto, es decir que se mantienen invariables sin importarle con cuales propiedades o cualidades se relacionen. [11]

R 1 Autenticar usuario.

1.1 El sistema debe mostrar a los usuarios la pantalla de autenticación.

1.1.1 El sistema debe permitir insertar usuario y contraseña.

1.2 Se validan los datos entrados.

1.1.2 Si el usuario no posee permisos para acceder, el sistema debe notificarlo e impedirle el acceso.

1.3 El sistema debe mostrar las opciones a las que el usuario tiene acceso y finaliza el caso de uso.

R 2 Gestionar solicitud diurna.

2.1 Nueva solicitud diurna.

2.1.1 El sistema debe mostrar un formulario con los campos de los datos específicos de una solicitud diurna y la posibilidad de introducir el listado de visitantes que desea para esa solicitud.

2.1.1.1 Si la visita ha sido registrada con anterioridad, puede seleccionarse en el listado de visitas anteriores o realizar la búsqueda mediante el número de carnet de identidad del visitante.

2.1.2 Se validan los datos entrados.

2.1.2.1 Si los datos entrados no son válidos o existen campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

2.1.3 El sistema debe notificar que se adicionó la solicitud diurna correctamente.

2.2 Modificar solicitud diurna.

2.2.1 El sistema debe mostrar un formulario con los campos de datos de la solicitud diurna y la posibilidad de modificarlos.

2.2.1.1 El sistema debe brindar la posibilidad de seleccionar visitantes registrados, mostrando un listado de visitas insertadas anteriormente o realizar la búsqueda mediante el número de carnet de identidad del visitante.

2.2.2 Se validan los datos entrados.

2.2.2.1 Si los datos entrados no son válidos o existen campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

2.2.3 El sistema debe notificar que se modificó la solicitud correctamente.

2.3 Cancelar solicitud diurna.

2.3.1 El sistema debe permitir eliminar cualquier solicitud que se encuentre en el listado, Enviadas, en espera de ser aprobada o denegada.

2.3.1.1 El sistema debe mostrar un mensaje de confirmación antes de eliminar.

2.3.2 El sistema debe notificar que la solicitud se eliminó correctamente y mostrar la página principal.

2.4 Detalles solicitud diurna.

2.4.1 El sistema debe mostrar todos los detalles de la solicitud en el listado Últimas Enviadas.

R 3 Gestionar solicitud temporal.

3.1 Nueva solicitud temporal.

3.1.1 El sistema debe mostrar un formulario con los campos de los datos específicos de una solicitud temporal y la posibilidad de introducir el listado de visitantes que desea para esa solicitud.

2.1.1.2 Si la visita ha sido registrada con anterioridad, puede seleccionarse en el listado de visitas anteriores o realizar la búsqueda mediante el número de carnet de identidad del visitante.

3.1.1 Se validan los datos entrados.

2.1.2.2 Si los datos entrados no son válidos o existen campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

3.1.2 El sistema debe notificar que se adicionó la solicitud temporal correctamente.

3.2 Modificar solicitud temporal.

3.2.1 El sistema debe mostrar un formulario con los campos de datos de la solicitud temporal y la posibilidad de modificarlos.

3.2.1.1 El sistema debe brindar la posibilidad de seleccionar visitantes registrados, mostrando un listado de visitas insertadas anteriormente o realizar la búsqueda mediante el número de carnet de identidad del visitante.

3.2.2 Se validan los datos entrados.

3.2.2.1 Si los datos entrados no son válidos o existes campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

3.2.3 El sistema debe notificar que se modificó la solicitud temporal correctamente.

3.3 Cancelar solicitud temporal.

3.3.1 El sistema debe permitir eliminar cualquier solicitud que se encuentre en el listado Enviadas, en espera de ser aprobada o denegada.

2.3.1.2 El sistema debe mostrar un mensaje de solicitud de confirmación antes de eliminar.

3.3.2 El sistema debe notificar que la solicitud se eliminó correctamente y mostrar la página principal.

3.4 Detalles solicitud diurna.

3.4.1 El sistema debe mostrar todos los detalles de la solicitud en el listado Últimas Enviadas.

R 4 Gestionar solicitud al hotelito.

4.1 Nueva solicitud al hotelito.

4.1.1 El sistema debe mostrar un formulario a llenar con los campos de datos de una solicitud al hotelito y debe dar la posibilidad al Directivo de crear una nueva.

4.1.1.1 El sistema debe brindar la posibilidad de seleccionar visitantes registrados, mostrando un listado de visitas insertadas anteriormente o realizar la búsqueda mediante el número de carnet de identidad del visitante.

4.1.2 Se validan los datos entrados.

4.1.2.1 Si los datos entrados no son válidos o existen campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

4.1.3 El sistema debe notificar que se adicionó la solicitud al hotelito correctamente.

4.2 Modificar solicitud al hotelito.

4.2.1 El sistema debe mostrar un formulario con los campos de datos de una solicitud al hotelito y la posibilidad de modificarlos.

4.2.1.1 El sistema debe brindar la posibilidad de seleccionar visitantes registrados, mostrando un listado de visitas insertadas anteriormente.

4.2.2 Se validan los datos entrados.

4.2.2.1 Si los datos entrados no son válidos o existes campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

4.2.3 El sistema debe notificar que se modificó la solicitud correctamente.

4.3 Cancelar solicitud al hotelito.

4.3.1 El sistema debe permitir eliminar cualquier solicitud al hotelito que se encuentre en espera de ser aprobada o denegada.

4.3.1.1 El sistema debe mostrar un mensaje de confirmación antes de eliminar.

4.3.2 El sistema debe notificar que la solicitud se eliminó correctamente.

4.4 Detalles solicitud al hotelito.

4.4.1 El sistema debe mostrar todos los detalles de la solicitud en el listado Últimas Enviadas.

R 5 Notificar visita.

5.1 El sistema debe permitir al Directivo seleccionar el tipo de visita que desea notificar (visita diurna, visita temporal, visita al hotelito).

5.2 El sistema debe mostrar un formulario con los campos correspondientes al tipo de solicitud seleccionada y dar la posibilidad de introducir los datos.

5.2.1 El sistema debe brindar la posibilidad de seleccionar visitantes registrados, mostrando un listado de visitas insertadas anteriormente o realizar la búsqueda mediante el número de carnet de identidad del visitante.

5.3 Se validan los datos entrados.

5.3.1 Si los datos entrados no son válidos o existen campos vacíos, se notifica mediante un mensaje y se da la posibilidad de modificarlos nuevamente.

5.4 El sistema debe notificar que se adicionó la visita correctamente.

R 6 Aprobar/denegar solicitudes.

6.1 Aprobar/Denegar solicitudes.

6.1.1 El sistema debe mostrar el listado de las solicitudes pendientes de aprobación correspondientes al área del Directivo.

6.1.2 Al seleccionar la solicitud debe brindar la posibilidad de aprobar o denegar la solicitud.

6.1.3 En caso de denegar la solicitud el sistema debe solicitar introducir los motivos de la negación.

6.1.4 El sistema debe notificar que se aprobó o denegó la solicitud correctamente.

6.1.5 Se actualiza el listado de solicitudes.

6.2 Detalles de solicitud.

6.2.1 El sistema debe mostrar todos los detalles de la solicitud en el listado Últimas Enviadas.

R 7 Registrar entrada/salida de visitas.

7.1 Registrar entrada/salida de visitas.

7.1.1 El sistema debe mostrar un formulario que posea como criterio de búsqueda en número de carnet de identidad del visitante.

7.1.2 El sistema deberá desplegar a partir de que el usuario introduzca los primeros tres números del CI el listado de las visitas que poseen esos números en su carnet de identidad.

7.1.2.1 En caso de no encontrar al visitante se debe notificar al mostrar el listado vacío y se da la posibilidad de introducir los datos nuevamente.

7.1.3 Se deben mostrar los detalles de la visita con la posibilidad de registrar su entrada.

7.1.4.1 En caso de que el visitante se encuentre abandonando el centro solo permitirá registrar la salida y solicitará introducir observaciones de la visita.

7.1.4 El sistema debe notificar que se registró correctamente la entrada/salida.

7.2 Registrar visita imprevista.

7.2.1 El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al Técnico SP de registrar una nueva visita imprevista.

7.2.2 Se validan los datos entrados.

7.2.2.1 Si los datos no son válidos o existen campos vacíos, se notifica mediante un mensaje y se da la posibilidad de introducirlos nuevamente.

7.2.3 El sistema debe notificar que se registró la visita correctamente.

R 8 Mostrar historial.

8.1 Mostrar historial.

8.1.1 El sistema debe mostrar el listado de todas las solicitudes realizadas por el usuario.

8.2 Detalles.

8.2.1 El sistema debe mostrar todos los detalles de la solicitud o visita.

R 9 Mostrar reporte SP.

9.1 El sistema debe mostrar diferentes criterios de búsqueda con la oportunidad de seleccionar datos en cualquiera de ellos.

9.2 El sistema debe mostrar los resultados de la búsqueda con la posibilidad de ver los detalles de aquellos encontrados.

R 10 Gestionar usuario.

10.1 Adicionar usuario.

10.1.1 El sistema debe mostrar un formulario y debe dar la posibilidad al administrador de crear un nuevo usuario.

10.1.2 El sistema debe mostrar un listado de permisos y roles con la posibilidad de asignarlos al usuario.

10.1.3 Se validan los datos entrados.

10.1.3.1 En caso de faltar rol o permiso por asignar al usuario, el sistema debe notificarlo mediante un mensaje.

- 10.1.4 Se almacena el usuario adicionado.
 - 10.1.4.1 Si el usuario ya existe se notifica mediante un mensaje y se da la posibilidad de crear el usuario nuevamente.
- 10.1.5 El sistema debe notificar que se creó el usuario correctamente.
- 10.2 Modificar usuario.
 - 10.2.1 El sistema debe mostrar un formulario con los datos del usuario y la posibilidad de cambiarlos.
 - 10.2.2 Se validan los datos.
 - 10.2.2.1 En caso de existir error en los cambios, se debe notificar mediante un mensaje.
 - 10.2.3 El sistema debe notificar que se modificaron los datos correctamente.
- 10.3 Eliminar usuario.
 - 10.3.1 El sistema debe permitir eliminar el usuario deseado.
 - 10.3.1.1 El sistema debe mostrar un mensaje de solicitud de confirmación antes de eliminar.
 - 10.3.2 El sistema debe notificar que el usuario se eliminó correctamente.

R 11 Gestionar rol.

- 11.1 Adicionar rol.
 - 11.1.1 El sistema debe mostrar un formulario y debe dar la posibilidad al administrador de insertar los datos del nuevo rol.
 - 11.1.2 El sistema debe mostrar el listado de usuarios y permisos que pueden ser asignados al rol con la posibilidad de seleccionarlos.
 - 11.1.3 Se validan los datos entrados.
 - 10.1.3.1 Si el rol ya existe, se notifica mediante un mensaje y se da la posibilidad de crear el rol nuevamente.
 - 11.1.4 El sistema debe notificar que se creó el rol correctamente.
- 11.2 Eliminar rol.
 - 11.2.1 El sistema debe permitir eliminar el rol deseado.
 - 11.2.2 El sistema debe mostrar un mensaje de solicitud de confirmación antes de eliminar.
 - 11.2.3 El sistema debe notificar que el rol se eliminó correctamente.

R 12 Gestionar permiso.

12.1 Adicionar permiso.

12.1.1 El sistema debe mostrar un formulario a llenar y debe dar la posibilidad al administrador de crear un permiso y el mismo ser asignado a roles o usuarios registrados en el sistema.

12.1.2 Se validan los datos entrados.

12.1.2.1 Si el permiso ya existe se notifica mediante un mensaje y se da la posibilidad de crear el permiso nuevamente.

12.1.3 El sistema debe notificar que se creó el permiso correctamente.

12.2 Eliminar permiso.

12.2.1 El sistema debe permitir eliminar el permiso deseado.

12.2.1.1 El sistema debe mostrar un mensaje de solicitud de confirmación antes de eliminar.

12.2.2 El sistema debe notificar que el permiso se eliminó correctamente.

R 13 Solicitar prórroga para visita temporal.

13.1 El sistema debe mostrar el listado de las visitas temporales realizadas por el usuario que se encuentran en curso con la posibilidad de solicitar prórroga para la seleccionada.

13.2 El sistema debe mostrar un formulario anterior con la posibilidad de insertar los nuevos datos.

13.3 Se validan los datos entrados.

13.3.1 Si los datos no son válidos o existen campos vacíos, se notifica mediante un mensaje y se brinda la posibilidad de llenarlos nuevamente.

13.4 El sistema debe notificar que se realizó la solicitud correctamente.

R 14 Solicitar prórroga para visita hotelito.

14.1 El sistema debe mostrar el listado de los eventos para los cuales el Directivo posee visitas en curso y la posibilidad de solicitar prórroga para la selecciona.

14.2 El sistema debe mostrar el listado de las visitas en curso que pertenecen al evento seleccionado y la posibilidad de seleccionar para los (el) que desea solicitar prórroga.

- 14.3 El sistema debe mostrar un formulario y la posibilidad de insertar la nueva fecha de salida y los motivos de la solicitud.
- 14.4 Se validan los datos.
 - 14.4.1 Si los datos no son válidos o existen campos vacíos, se notifica mediante un mensaje y se brinda la posibilidad de llenarlos nuevamente.
- 14.5 Si los datos no son válidos o existen campos vacíos, se notifica mediante un mensaje y se brinda la posibilidad de llenarlos nuevamente.
- 14.6 El sistema debe notificar que se realizó la solicitud de prórroga de visita al hotelito correctamente.

2.7.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas características son las que permiten al producto ser atractivo, usable, rápido y confiable. [11]

➤ Requerimientos de software

En las máquinas clientes sólo se requiere de un navegador web, preferentemente Internet Explorer (6.0 o superior) o Mozilla Firefox (2.0 o superior). Se requiere que el servidor (host) tenga instalado PostgresSql 8.3, Apache 2.0 y PHP 5. El sistema operativo puede ser cualquiera que soporte estas herramientas.

➤ Requerimientos de hardware

En el cliente se requiere, al menos una computadora que contenga como mínimo 128 MB de RAM, el servidor debe tener 256 MB de RAM como mínimo, todas las PC en las que se use la aplicación deben estar conectadas a la red local.

➤ Restricciones en el diseño y la implementación.

Todas las herramientas usadas en el desarrollo del sistema deben ser libres, o en caso de no serlos que la universidad haya adquirido su licencia. Como metodología de desarrollo se utilizará RUP, generando sólo los artefactos imprescindibles, para lograr el desarrollo ágil del software.

➤ Requerimientos de apariencia o interfaz externa

La interfaz debe ser amigable, intuitiva y fácil de usar, permitiendo que personas con pocas habilidades informáticas manejen el sistema.

➤ Requerimientos de Seguridad

Existencia de distintos roles que establezcan que la información solo sea vista por aquellos usuarios que posean los privilegios suficientes; restringir la ejecución de acciones por usuarios sin privilegios a acceder a las mismas y se debe realizar la verificación sobre acciones irreversibles como eliminaciones.

➤ Requerimientos de Usabilidad

La aplicación deberá ser usada por aquellas personas con las que tiene relación la información que se maneja. Esto incumbe principalmente a trabajadores de la universidad. La aplicación debe garantizar un acceso fácil y rápido, contando con un menú que satisfaga las necesidades de los usuarios.

➤ Requerimientos de Soporte

Garantía de instalación y prueba del sistema, además de un breve entrenamiento a los futuros usuarios.

➤ Requerimientos de Confiabilidad

Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

➤ Requerimientos de Rendimiento

Los tiempos de respuestas deben ser los más rápidos al igual que la velocidad de procesamiento de la información.

➤ Requerimientos de Disponibilidad.

Se debe garantizar el funcionamiento de la aplicación las 24 horas al día los siete días de la semana con el menor tiempo posible de recuperación de fallos. Se deben crear copias de respaldo periódicas que puedan restaurar el sistema en caso de fallo crítico o pérdida total de la información.

➤ **Requerimientos de Persistencia.**

La información del sistema debe almacenarse en bases de datos con carácter imborrable con el objetivo de poder realizar análisis de la misma en cualquier momento durante el paso de los años.

➤ **Requerimientos Legales.**

Las herramientas y las tecnologías en que estará basada la aplicación informática deberán cumplir con las licencias de software libre.

2.8 Modelo de casos de uso del sistema

2.8.1 Definición de los actores del sistema a automatizar

Nombre del Usuario	Descripción
Usuario.	Toda persona con permisos para acceder al sistema.
Administrador.	Usuario con todos los privilegios en el sistema.
Directivo.	Responsable de un área determinada.
Trabajador UCI.	Trabajador de un área específica de la universidad con permisos para acceder al sistema.
Trabajador Interno.	Especificación de Trabajador UCI que comprende aquellos trabajadores que residen en la universidad.
Técnico Seguridad y Protección.	Encargado de registrar el acceso de personal a la universidad.
Especialista de Seguridad y Protección.	Trabajador que radica en la oficina de Seguridad y Protección, el cual es el encargado, entre otras cosas, del control de todas las vistas que se realizan a la universidad.

Tabla 2.6 Descripción de actores del sistema.

2.8.2 Diagramas de casos de uso a automatizar

Un diagrama de casos de uso del sistema, muestra qué actores interactúan con qué casos de uso. [11]

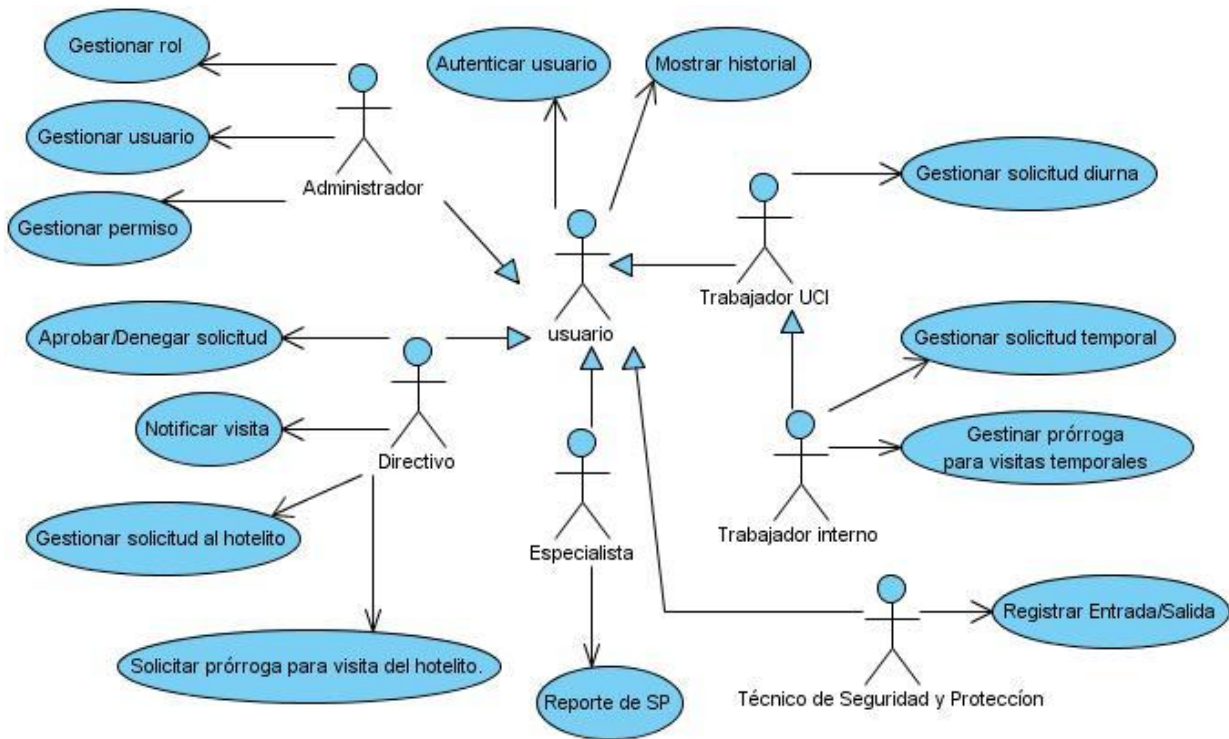


Figura 8 Diagrama de Casos de Uso del Sistema.

2.8.3 Descripción textual de los casos de uso del sistema

A través de la descripción textual de los casos de uso del sistema se puede detallar paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema, convirtiéndose este en el último paso para transitar a la construcción de la solución propuesta.

En el anexo #3 se muestran las descripciones de los casos de uso del sistema, observándose la interacción actor-sistema.

2.9 Estimación del esfuerzo

2.9.1 Planificación por Casos de Uso

“La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema. Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas como el Proceso Unificado de Rational (Rational Unified Process)” [10]

Una vez determinados los casos de uso, el análisis de Puntos de Casos de Uso es el método de estimación del tiempo de desarrollo más apropiado para contabilizar el tiempo total estimado de duración para el proyecto.

A continuación se definen los pasos a seguir para el desarrollo de este método.

2.9.2 Paso 1 Cálculo de Puntos de Casos de Uso sin Ajustar

La ecuación $UUCP = UAW + UUCW$ responde al cálculo de Puntos de Casos de Uso sin ajustar.

En la que:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

2.9.3 Factor de Peso de Actores sin ajustar

El Factor de Peso de los Actores sin ajustar (*UAW*) se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos [12] como se muestra a continuación:

Tipo de Actor	Descripción	Factor de Peso	Actores	Total
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0	0
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0

Complejo	Una persona que interactúa con el sistema mediante una interfaz grafica.	3	9	23
Total				23

Tabla 2.7 Factor de Peso de los Actores sin Ajustar.

En el sistema propuesto poseerá la interacción de 9 actores (Usuario, Administrador, Directivo, Trabajador UCI, Trabajador Interno, Técnico Seguridad y Protección, Servicios, Instructora, Especialista de Seguridad y Protección) los cuales representan actores complejos para el sistema, ya que interactúan a través de una interfaz grafica con el mismo, a los que se le asigna un peso de 3. Pudiendo calcular así el factor de peso de los actores sin ajustar mediante la ecuación:

$$UAW = \Sigma \text{cantidad de actores} * \text{peso}$$

$$UAW = 23$$

2.9.4 Factor de Peso de Casos de Uso sin Ajustar

A través de un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos se calcula el valor de Factor de Peso de los Casos de Uso sin ajustar [12]. La complejidad de los Casos de Uso se crea teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, dado que en una transacción se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia como se muestra a continuación en la siguiente tabla:

Tipos de Casos de Uso	Descripción	Factor de Peso	Casos de Uso	Total
Simple	El caso de uso contiene de 1 a 3 transacciones.	5	8	40
Medio	El caso de uso contiene de 4 a 3 transacciones.	10	4	40
Complejo	El caso de uso contiene 8 o más transacciones.	15	4	60
Total				140

Tabla 2.8 Factor de Peso de Casos de Uso sin Ajustar.

Se hallan 8 casos de uso que contienen de 1 a 3 transacciones (de tipo simple), 4 casos de uso que contienen de 4 a 3 transacciones (de tipo medio) y 4 que contienen 8 o más transacciones (de tipo complejo) quedando el factor de casos de uso sin ajustar:

$$UUCW = \Sigma \text{ cantidad de casos de uso } * \text{ peso}$$

$$UUCW = 140$$

Implicando finalmente que los Puntos de Casos de Uso sin ajustar sean:

$$UUCP = UAW + UUCW$$

$$UUCP = 23 + 140$$

$$UUCP = 163$$

2.9.5 Paso 2. Cálculo de los puntos de casos de uso ajustados

Para ajustar el valor del Cálculo de los Puntos de Casos de Uso sin ajustar se deberá utilizar la ecuación:

$$UCP = UUCP * TCF * EF$$

En la que:

UCP: Puntos de Casos de Uso Ajustados.

UUCP: Puntos de Casos de Uso Sin Ajustar.

TCF: Factor de Complejidad Técnica

EF: Factor de Ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de 13 factores que evalúan la complejidad técnica del sistema. Cada uno de estos factores tienen un peso definido con los cuales se obtendrá puntos ponderados por cada uno de ellos, según la valoración que se le asigne la cual se cuantifica en un valor desde 0-2 (aporte irrelevante), 3-4

(aporte medio) , 5 (aporte muy relevante) [12]. En la tabla 2.26 se muestra el peso de cada uno de estos factores acompañado de una breve descripción así como su valor asignado:

Factor	Descripción	Peso	Valor Asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	3	3
T3	Eficiencia del usuario final	1	2	2
T4	Funcionamiento Interno complejo	1	2	2
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0,5	2	1
T3	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	1	2
T9	Facilidad de cambio	1	2	2
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	3	3
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	3	3
Total				28

Tabla 2.9 Factor de complejidad técnica.

La ecuación para este punto es:

$$TCF = 0.6 + 0.01 * \sum (\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * 28$$

$$TCF = 0.88$$

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente (EF). Se trata de un conjunto de 8 factores que se cuantifican con valores de 0 (aporte irrelevante) a 5 (aporte relevante) [12]. En la siguiente tabla se asignan los pesos correspondientes al sistema que se propone acompañados de un breve comentario:

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	4.5
E2	Experiencia en la aplicación.	0,5	2	1
E3	Experiencia en la orientación a objetivos.	1	2	2
E4	Capacidad del analista líder.	0,5	4	2
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos.	2	4	8
E3	Personal Part-Time.	-1	0	0
E8	Dificultad del lenguaje de programación.	-1	1	1
Total				21.5

Tabla 2.10 Factor del Factor Ambiente

$$EF = 1.4 - 0.03 * \Sigma (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * 21.5$$

$$EF = 0.355$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 163 * 0.88 * 0.355$$

$$UCP = 110.9548$$

2.9.6 Paso3. Estimación de esfuerzo a través de puntos de casos de uso

$$E = UCP * CF$$

En la que:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para alcanzar el factor de conversión CF:

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6. Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E3 y E8.

- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.

- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto. [12]

En este caso se puede decir que:

$$CF = 20 \text{ Horas-Hombre} / \text{Punto de Casos de uso.}$$

$$CF = 110.9548 * 20$$

$$CF = 2219.096 \text{ Horas-Hombre.}$$

2.9.7 Paso 4. Calcular el esfuerzo de todo el proyecto

La siguiente tabla se muestra el esfuerzo total Horas-Hombre

Actividad	Porcentaje %	Horas- Hombres
Análisis	10	515,8
Diseño	20	1031,6
Implementación	40	2063,2
Pruebas	15	333,3
Sobrecarga (otras actividades)	15	333,3
Total	100	5158

Tabla 2.11 Esfuerzo en Horas-Hombre.

El presente trabajo abarcará los flujos de trabajo análisis, diseño, implementación y prueba por lo que $ET = 2812$.

Suponiendo que una cada integrante trabaje 8 horas por día, y un mes tiene como promedio 30 días de los cuales solo serían laborables 26; la cantidad de horas que puede trabajar una persona en 1 mes es 208 horas, a partir de lo cual :

$ET = 13.5$ mes – hombre.

Luego, teniendo en cuenta que:

Tiempo = ET / CH

Tiempo = $13.5 / 2$

Tiempo = 6.35

A partir de este análisis se concluye que con 2 hombres trabajando en la realización del análisis, diseño, implementación y prueba, el mismo se desarrolla en un periodo de 6 meses y medio.

2.9.8 Costo del proyecto

El salario promedio mensual es de 100\$

CH: Cantidad de hombres = 2

CHM = 2* Salario promedio

CHM = $2 * 100\$$

CHM = 200\$/mes

Costo = CHM * ET / CH

Costo = $200 * 13.5/2$

Costo = 2300\$

2.10 Beneficios tangibles e intangibles

Su principal objetivo es ayudar al control de la seguridad interna de la UCI, específicamente la seguridad referida a las visitas que se realizan diariamente.

Por tanto, los beneficios inmediatos son mayormente intangibles:

1. Facilidad y seguridad en el método de reservación y autorización de las solicitudes.
2. Ahorro de tiempo en la búsqueda de información de cualquier visitante que se encuentre o haya visitado la UCI.
3. Control de cada visita que accede a la UCI, con fecha y hora, lugar, motivos, solicitante, etc.
4. Posibilidad de ver reportes de accesos con facilidad de búsqueda.
5. Aumento de la seguridad interna de la UCI

2.11 Análisis de Costo y Beneficios

El desarrollo del Módulo de Visitas para el Sistema de Control de Acceso, no requiere de grandes gastos de recursos, ni de tiempo. La base de datos que almacena la información, puede ser insertada en los servidores de la universidad.

La tecnología utilizada para el desarrollo del sistema es en su totalidad software libre, lo que significa que son de gratis acceso. Entre las características de la utilización de este tipo de herramientas se encuentra el ahorro en costos de adquisición, de implementación, de soporte o mantenimiento, y de interoperabilidad.

Se realizó el estudio de otras soluciones de control de acceso de visitas disponibles en el mundo, y se llegó a la conclusión de que no se ajustan a las características del entorno de la universidad, además, son sistemas demasiado costosos.

2.12 Conclusiones

Este capítulo ha servido para desglosar los procesos del negocio y detallarlo en las descripciones textuales para su mejor comprensión. Se aplicó el método de la observación el constatar directamente y en la práctica como se desarrolla el negocio. Se analizaron y valoraron todos los requisitos necesarios para la implantación de un sistema que diera fin a la situación problemática actual. Se construyeron todos los artefactos propuestos por la metodología de desarrollo del software que, además de servir como guía en estos procesos, serán los

cimientos de las próximas fases de Análisis, Diseño e Implementación de la propuesta. Se demostró la factibilidad de la implementación del Módulo de Visitas.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se expondrán los artefactos resultantes de la fase de análisis y diseño siguiendo la metodología de desarrollo RUP. Se mostrarán los diagramas de clases del análisis y del diseño, así como el diseño de la Base de Datos y la descripción de sus tablas, que ayudarán a lograr una implementación acorde con las necesidades del usuario. Se presentaran también los patrones de diseño y arquitectura usados para el trabajo con el framework Symfony. Se abordara además todo lo referente al tratamiento de errores y la seguridad de la aplicación.

3.2 Análisis

Según Roger S. Pressman en su libro “Ingeniería de Software. Un enfoque Práctico” a partir de los años 80 se desarrollan un gran número de métodos de modelado, sin embargo todos se relacionan por un conjunto de principios operativos los cuales se mencionan a continuación:

1. Debe representarse y entenderse el dominio de información de un problema.
2. Deben definirse las funciones que se realizarán con el software.
3. Debe representarse el comportamiento del software (como consecuencia de acontecimientos externos).
4. Deben dividirse los modelos que representan la información, función y comportamiento de tal manera que se descubran los detalles por capas (o jerárquicamente).
5. El proceso de análisis debe ir desde la información esencial hasta el detalle de implementación.

Aplicando estos principios durante el análisis se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que posibilite ayuda a la hora de estructurar el sistema entero, incluyendo su arquitectura.

3.2.1 Diagrama de clases del análisis

Durante el análisis del sistema, el diagrama de clases del análisis se desarrolla en busca de una solución ideal, donde una clase representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema.

Las clases de análisis se fundamentan en tres estereotipos básicos: entidad, interfaz y control.

Clase entidad: Modelan información que posee larga vida y que es a menudo persistente.

Clase Interfaz: Modelan la interacción entre el sistema y sus actores.

Clase control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso. [13]

A continuación se describen los diagramas de clases del análisis de los casos de uso más significativos que conforman la solución propuesta. El resto de los diagramas podrán ser vistos en el anexo # 4

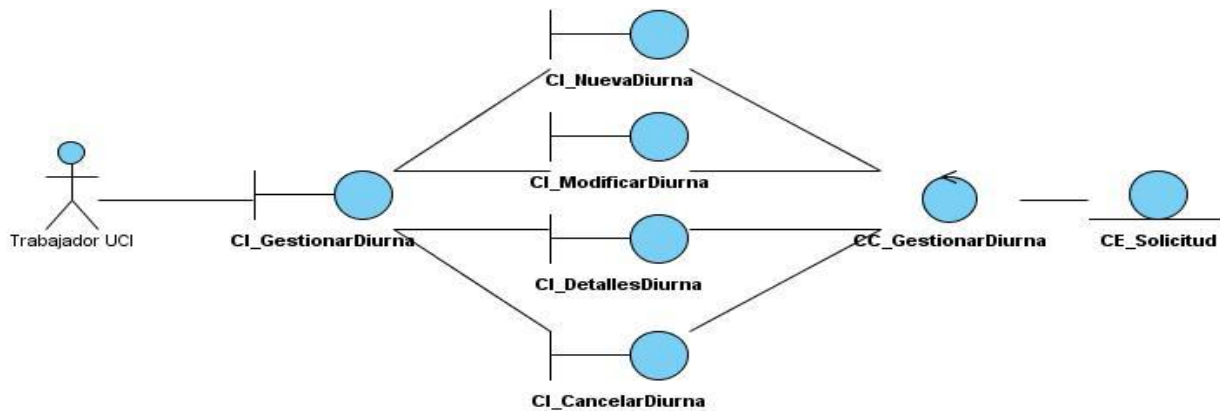


Figura 9 Diagrama de clases del análisis CU: Gestionar visita diurna.

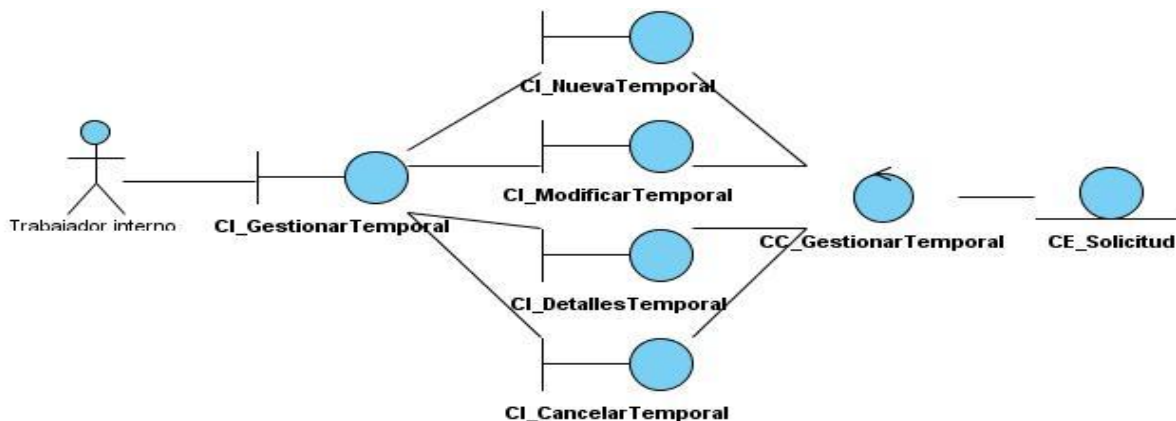


Figura 10 Diagrama de clases del análisis CU: Gestionar visita temporal.

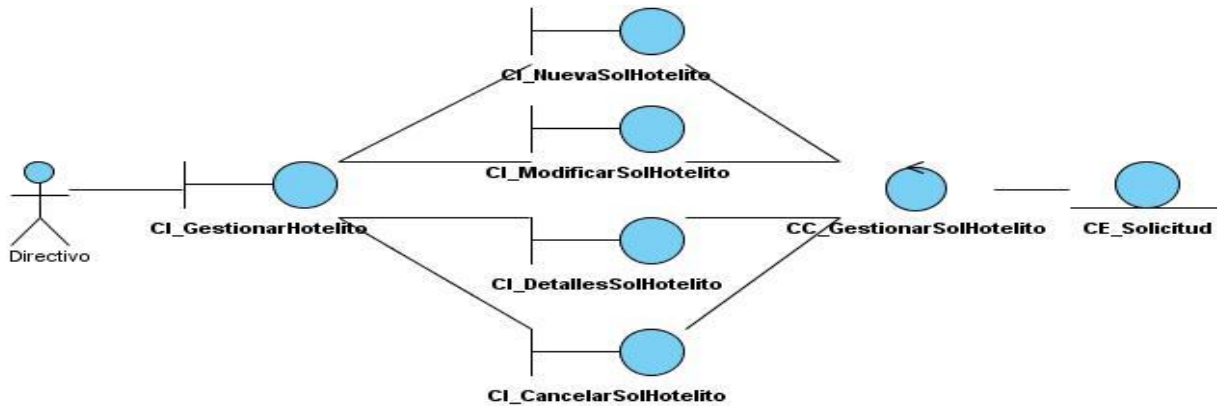


Figura 11 Diagrama de clases del análisis CU: Gestionar visita al hotelito.



Figura 12 Diagrama de clases del análisis CU: Aprobar/Denegar solicitudes.

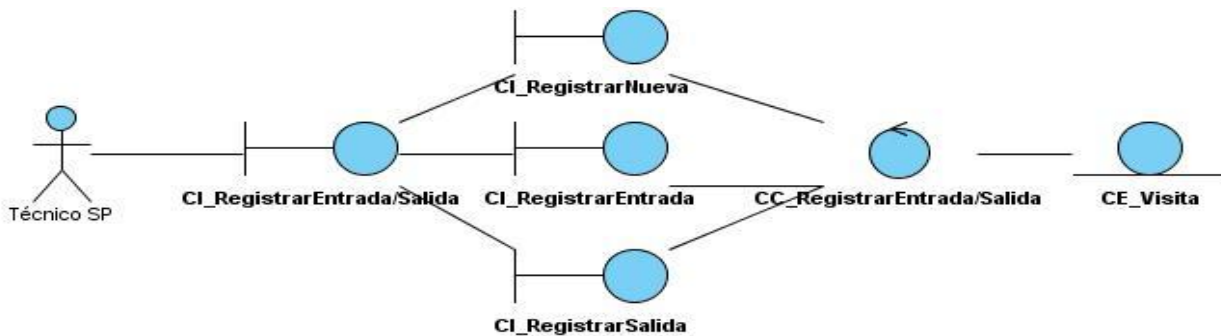


Figura 13 Diagrama de clases del análisis CU: Registrar entrada/salida de visitantes.

3.3 Diseño

El diseño es el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física.

Con el diseño se pretende construir un sistema que satisfaga determinadas especificaciones y respete requisitos sobre forma, rendimiento, utilización de recursos, coste, etc.

Los principios básicos del diseño plantean que:

- El diseñador debe considerar enfoques alternativos juzgando a cada uno en base a los requisitos del problema, los resultados disponibles y los criterios de calidad interna.
- Se deberían poder seguir los pasos de diseño hasta el modelo de análisis.
- El diseño no va a reinventar nada que ya esté inventado.
- El diseño debería presentar uniformidad e integración.
- Debe estructurarse para admitir cambios.
- El diseño no es escribir código y escribir código no es diseñar.
- Se debería valorar la calidad del diseño mientras se crea, no después de terminado. [13]

3.3.1 Diagrama de clases web del diseño

Este diagrama muestra la estructura de las clases que luego serán descritas en un lenguaje de programación las cuales satisfacen los detalles de la implementación. A continuación se muestran los diagramas de clases web del diseño de los casos de uso significativos del sistema. El resto de los diagramas podrán ser vistos en el anexo #5.

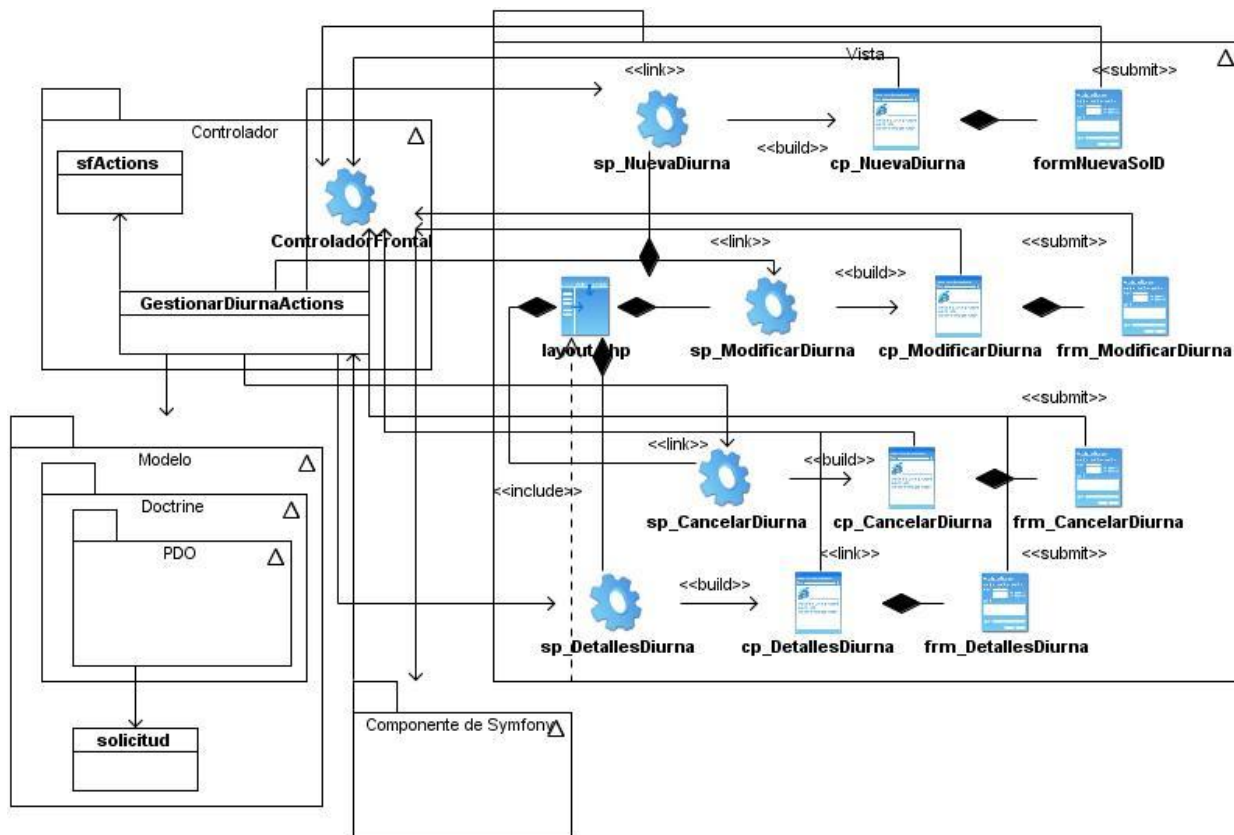


Figura 14 Diagrama de clases del diseño del CU Gestionar visita diurna.

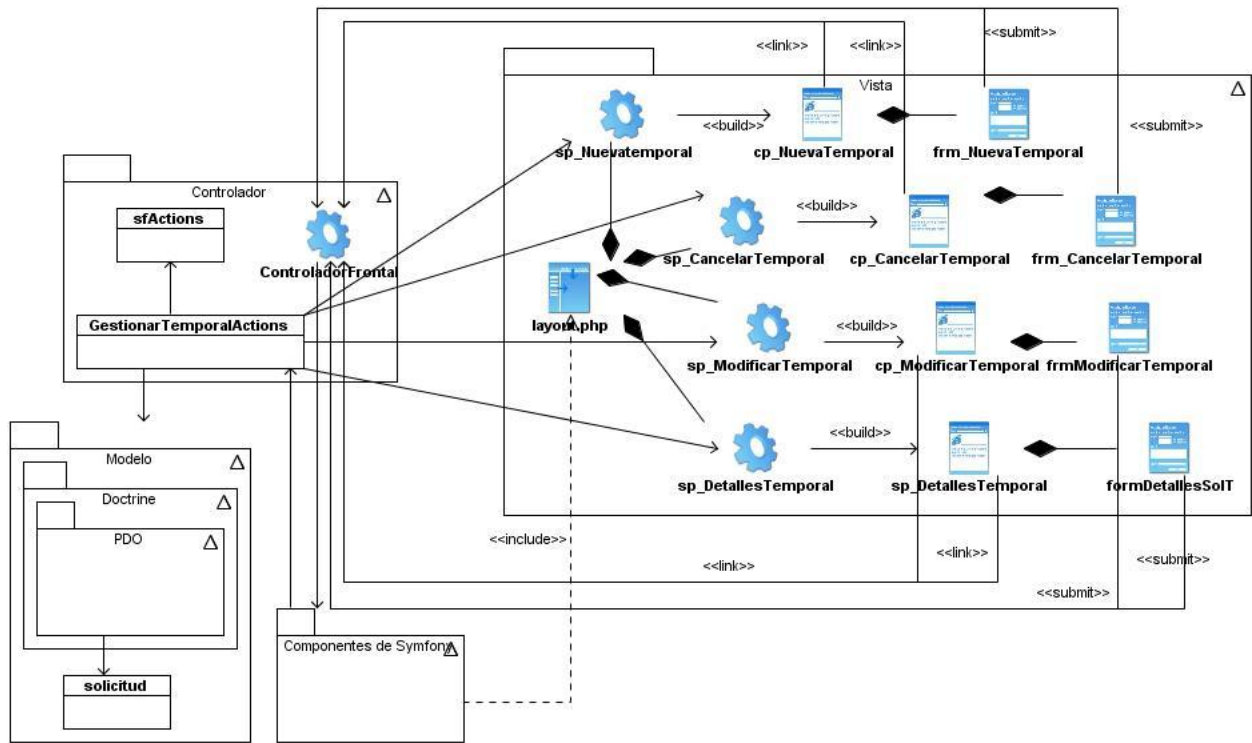


Figura 15 Diagrama de clases del diseño del CU Gestionar visita temporal.

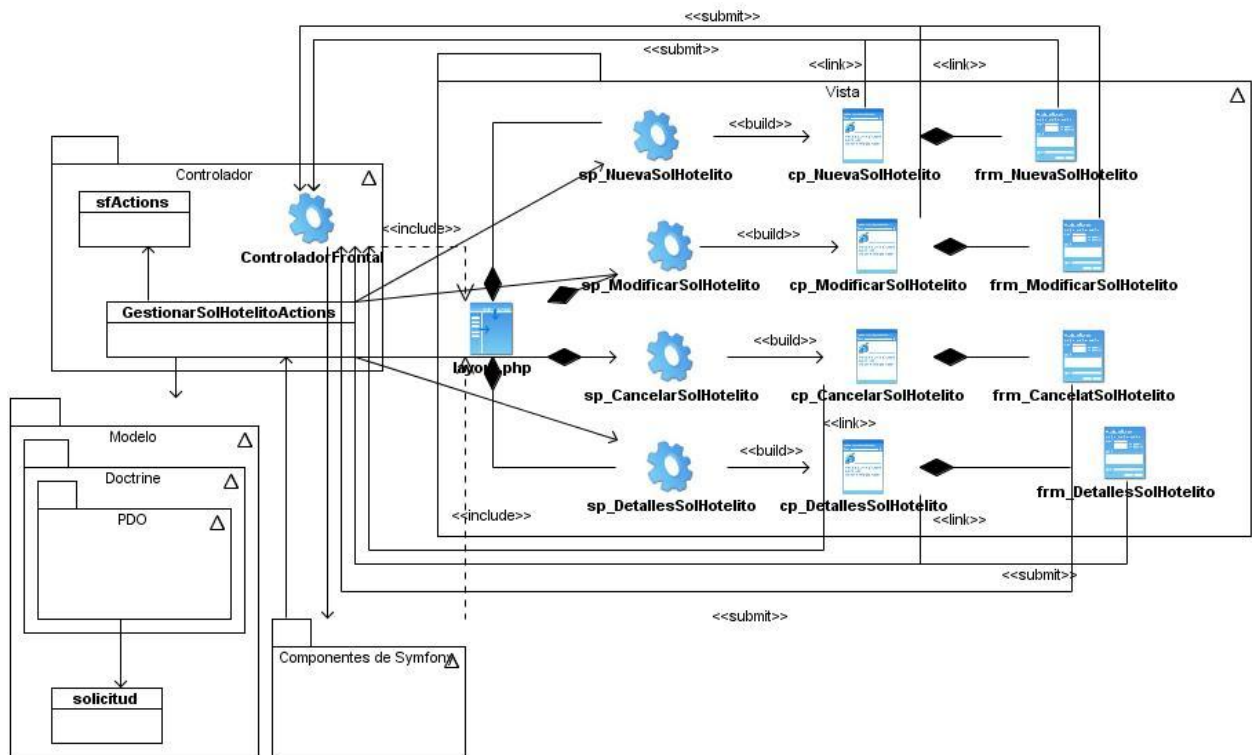


Figura 16 Diagrama de clases del diseño del CU Gestionar visita al hotelito.

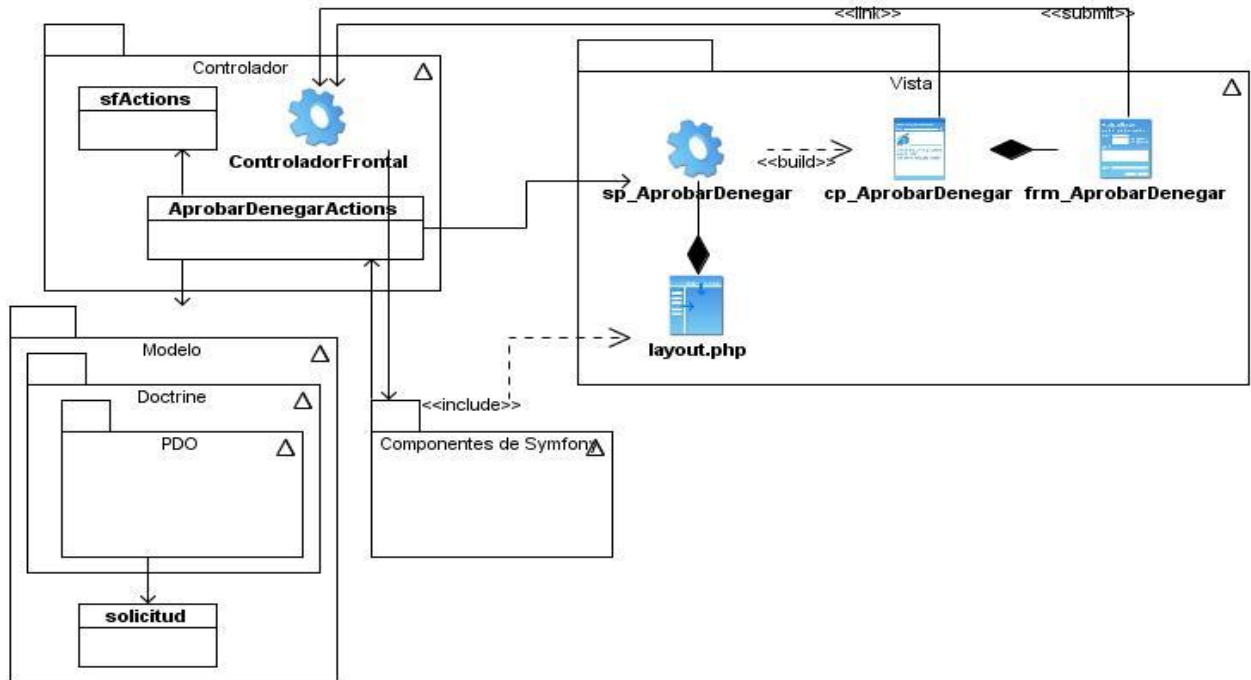


Figura 17 Diagrama de clases del diseño del CU Aprobar/denegar solicitudes.

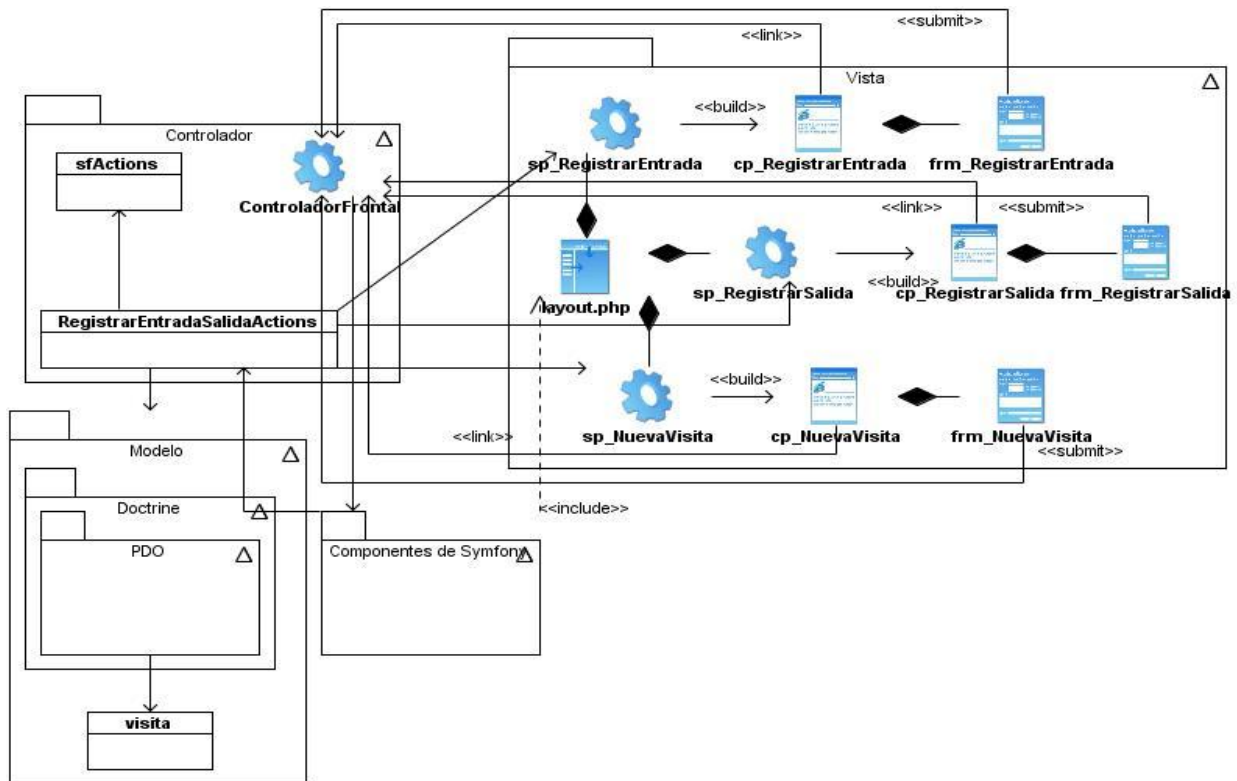


Figura 18 Diagrama de clases del diseño del CU Registrar entrada/salida de visitantes.

3.3.2 Descripción de las clases web del diseño

La descripción de estas clases queda plasmada en el anexo #6.

3.3.3 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

A continuación se muestran los diagramas de secuencia del caso de uso Gestionar visita diurna. El resto de los diagramas podrán verse en el anexo #7.

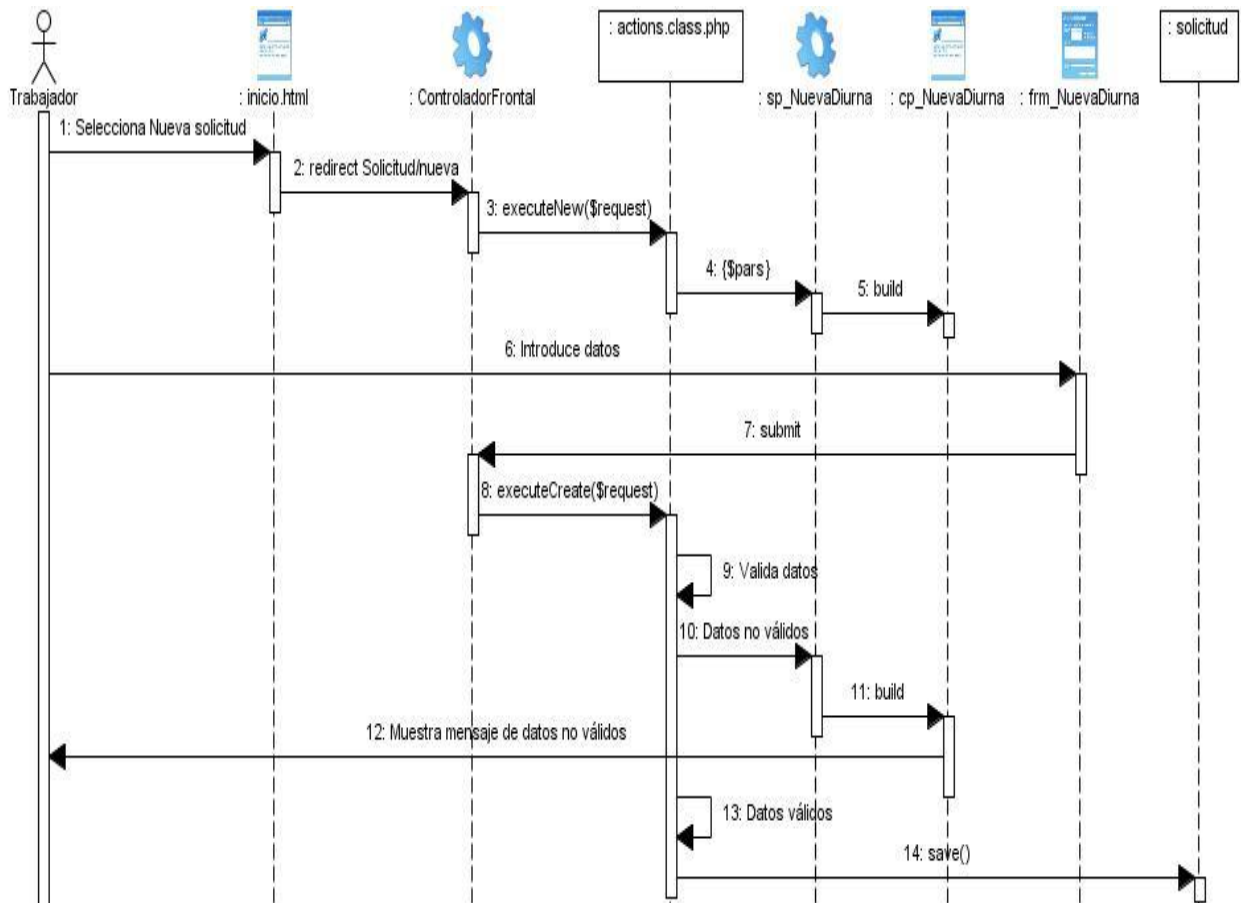


Figura 19 Diagrama de secuencia CU Gestionar visita diurna. Escenario: Nueva solicitud diurna.

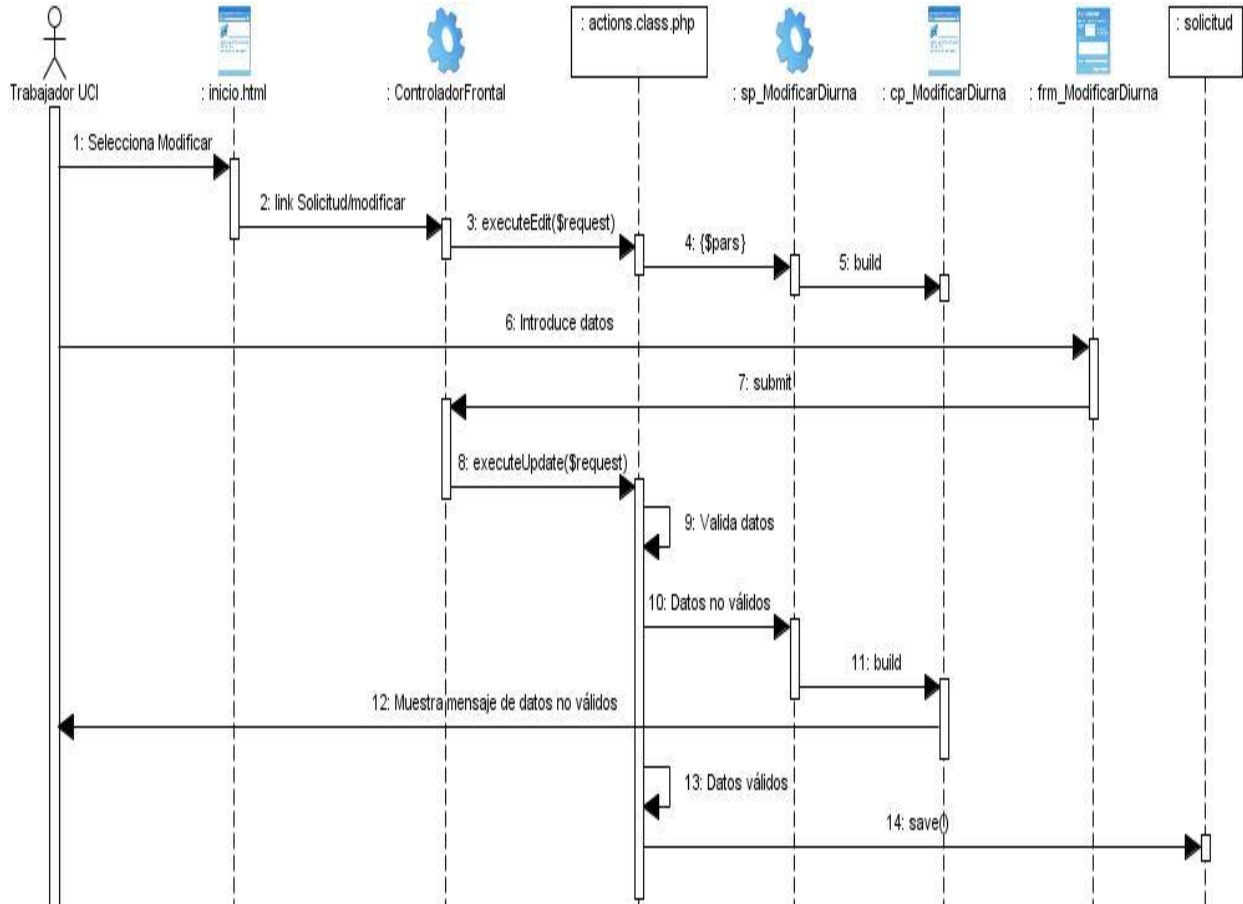


Figura 20 Diagrama de secuencia CU Gestionar visita diaria. Escenario: Modificar solicitud diaria.

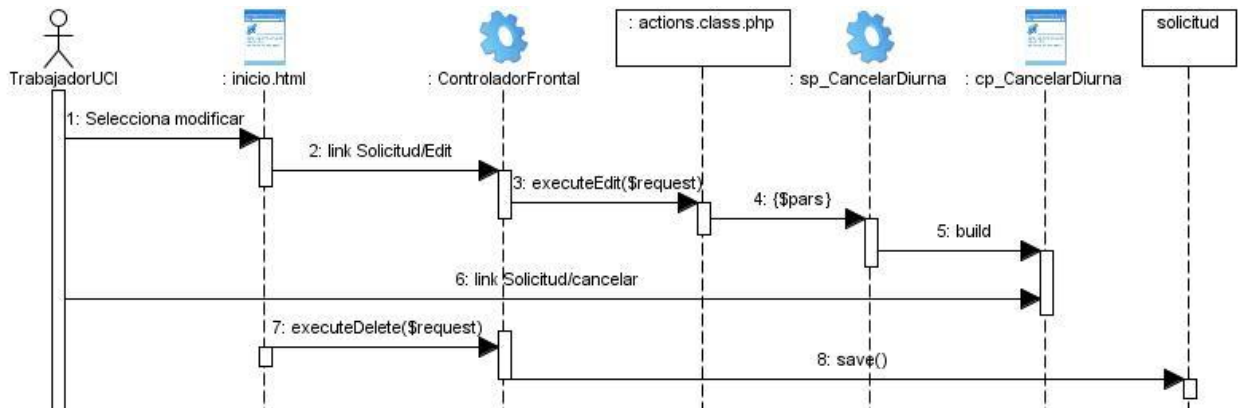


Figura 21 Diagrama de secuencia CU Gestionar visita diaria. Escenario: Cancelar solicitud diaria.

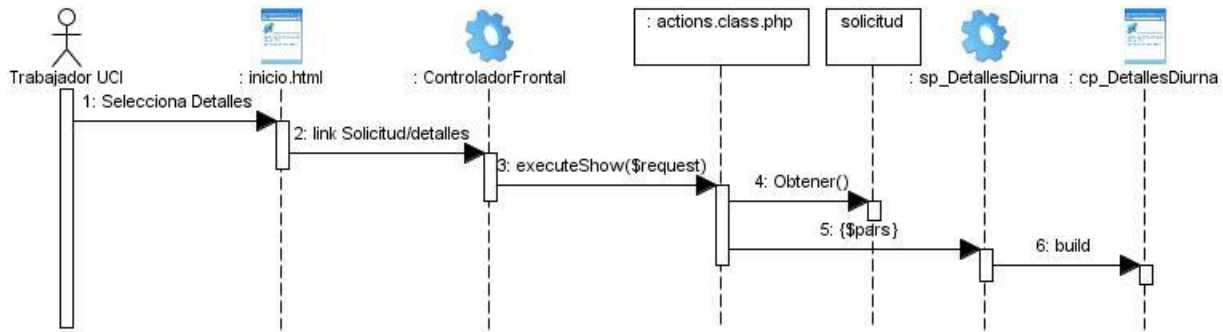


Figura 22 Diagrama de secuencia CU Gestionar visita diurna. Escenario: Detalles solicitud diurna.

3.4 Patrones de diseño y de arquitectura que implementa Symfony

3.4.1 Patrones aplicados en el diseño

El utilizar Symfony brinda la posibilidad de utilizar muchos patrones GRASP (Patrón de Asignación de Responsabilidades), a continuación se muestran algunos de los acreditados, los cuales serán ubicados en las capas de Modelo y Control que plantea el patrón arquitectónico MVC.

- **Patrón Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada [14]. Este patrón es muy utilizado al trabajar con Symfony ya que con la librería Doctrine para mapear la base de datos, se evidencia la continua utilización del mismo. Doctrine genera las clases para la gestión de las entidades con las responsabilidades fijadas según lo plantea el patrón Experto.
- **Patrón Creador:** Se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando: B contiene a A, B es una agregación (o composición) de A, B almacena a A, B tiene los datos de inicialización de A (datos que requiere su constructor) B usa a A. [14] En la clase actions.class.php se encuentran las acciones específicas para el módulo visita, solicitante, visitante, área y visitante. En cada acción se crean los objetos de las clases que son los que instancian a las clases del modelo, por lo que se demuestra que la clase actions.class.php es la “creadora” de estas entidades. Un ejemplo del uso de este patrón es en la acción executeCreate(\$request) que crea instancias de las clases entidades que contienen los datos a utilizar.

- **Patrón Alta Cohesión:** Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable [14] manteniendo la complejidad dentro de los límites manejables. El utilizar Symfony, permite crear y trabajar con clases con una alta cohesión ya que posibilita organizar el trabajo de acuerdo a la estructura del proyecto. Un ejemplo claro es la clase `actions.class.php`, ya que posee funcionalidades que poseen una fuerte relación, teniendo un sentido común y un único propósito, a la vez de que son las responsables de controlar las acciones de las plantillas.
- **Patrón Bajo Acoplamiento:** Debe haber pocas dependencias entre las clases [14] para poder extraerlas de un modo independiente y reutilizarlas. Este patrón se pone de manifiesto a partir de que el modelo posee las clases que implementan la lógica de negocio y acceso a datos, y estas clases no tienen relaciones con las de la vista o el controlador por lo que la dependencia es baja.
- **Patrón Controlador:** Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión [14]. Este patrón se pone de manifiesto en las clases `sfFrontController`, `sfWebFrontController`, `_sfContext`, los “actions” y el `index.php` del ambiente. La arquitectura del framework (MVC) brinda una capa específicamente para los controladores, que son el núcleo del mismo y especifica la presencia de este patrón.

3.4.2 Otros patrones aplicados en el diseño

Symfony también utiliza otros patrones de diseño, como son los patrones GoF. Entre los que se encuentran:

- **Patrón Singleton:** Este es un patrón de creación que es utilizado en la clase `sfRouting` utilizada por el controlador frontal (`sfWebFrontController`), esta clase se encarga de enrutar todas las peticiones que se realizan a la aplicación, por lo que es muy utilizada.
- **Patrón Command:** Este es un patrón de comportamiento. Se demuestra el uso de este patrón en la clase `sfWebFrontController`, específicamente en el método `dispatch()`, el cuál es el encargado de determinar cual modulo y acción utilizar en dependencia de la solicitud realizada por el usuario.
- **Patrón Decorador:** Patrón estructural que se pone de manifiesto a través del archivo `Layout.php` que es el que contiene el Layout de las páginas. Este archivo posee el código HTML que es común en todas las páginas de la aplicación, por lo que también es conocido

como plantilla global. Finalmente el layout es el que decora cada plantilla y este comportamiento es una implementación de este patrón.

➤ **Patrón Registry:** Este patrón es un medio simple y eficiente de compartir datos y objetos en la aplicación sin tener que preocuparse de mantener numerosos parámetros o hacer uso de variables globales, por lo que brinda muchos beneficios a los programadores en la POO. Se pone de manifiesto en la clase sfConfig, la cual se encarga de almacenar todas las variables de uso global en la aplicación.

3.4.3 Utilización del patrón que implementa Symfony

Symfony está basado en el patrón arquitectónico conocido como MVC, el cuál que está formado por tres niveles:

- El modelo, que representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista, que transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador, que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Esta arquitectura separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. La siguiente figura ilustra el funcionamiento del patrón MVC.

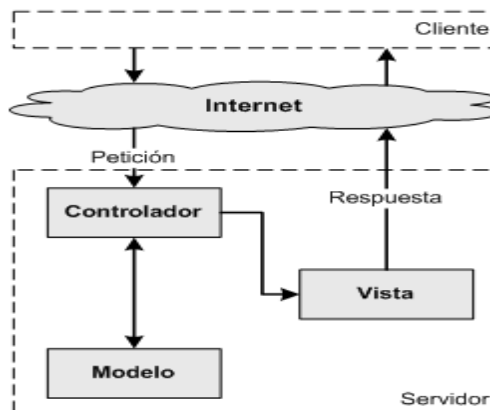


Figura 23 El patrón MVC.

Symfony realiza una implementación del MVC, toma lo mejor de esta arquitectura haciendo el desarrollo de las aplicaciones más rápido y sencillo [15]. Esta arquitectura obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework. El código de la presentación se guarda en la vista, el código de manipulación de datos en el modelo y la lógica de procesamiento de las peticiones constituye el controlador.

La vista se encarga de producir las páginas que se muestran como resultado de las acciones. Aquí se encuentran el layout y el controlador frontal, los cuales son comunes para todas las acciones de la aplicación, y este último es un componente que solo tiene código relativo al MVC, por lo que Symfony lo genera automáticamente.

En el controlador se encuentran las acciones, siendo estas el centro fundamental de la aplicación, ya que contienen toda la lógica de la aplicación. Las acciones utilizan el modelo y definen variables para la vista.

El modelo posee las clases que se generan automáticamente dependiendo de la base de datos. En Symfony este acceso a datos se realiza mediante objetos. La librería Doctrine crea la estructura básica de las clases y genera automáticamente el código necesario.

La abstracción de la base de datos es completamente invisible al programador, ya que la realiza otro componente específico llamado PDO. Así, si se cambia el sistema gestor de base de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración.

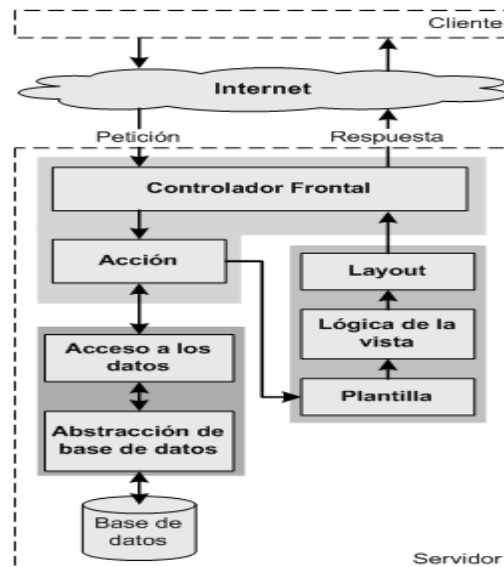
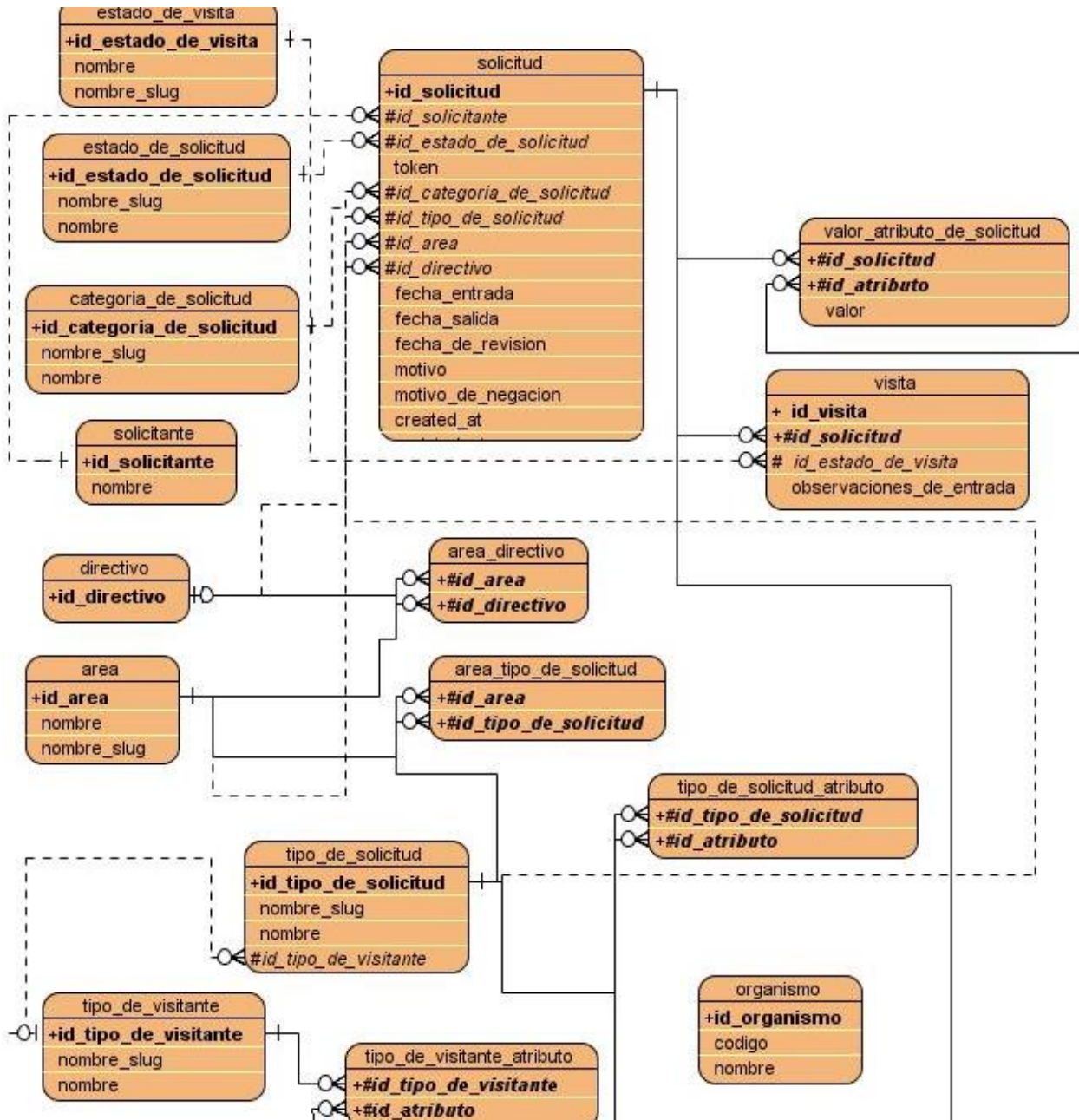


Figura 24 Flujo de trabajo de Symfony.

3.5 Diseño de la base de datos

A partir del diseño de la base de datos se representan las tablas de la misma así como las descripciones de cada uno con sus atributos.

3.5.1 Modelo entidad relación de la base de datos



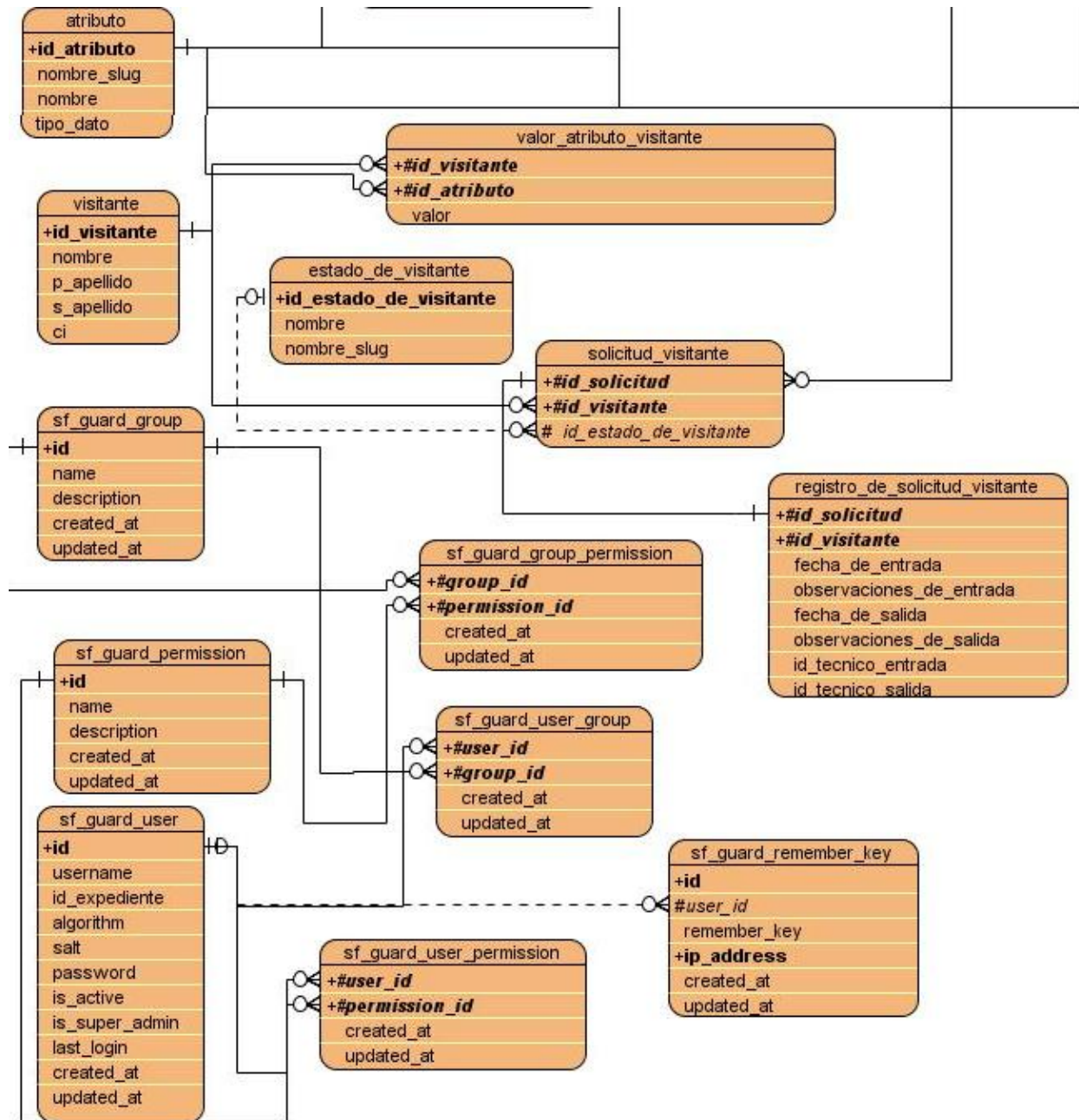


Figura 25 Modelo Entidad Relación.

3.5.2 Descripción de las tablas de la base de datos

Nombre	estado_de_visita.	
Descripción	En esta tabla se almacenan los posibles estados de las visitas los cuáles pueden ser: <i>pendiente</i> , <i>en curso</i> o <i>terminada</i> .	
Atributo	Tipo	Descripción

id_estado_de_visita	serial	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
nombre	character	Este campo almacena el nombre de los estados de la visita.
Nombre_slug	character	Nombre único que diferencia los elementos.

Tabla 3.1 Descripción de la tabla: estado_de_visita.

Nombre	estado_de_visitante	
Descripción	En esta tabla se almacenan los posibles estados de los visitantes los cuáles pueden ser: pendiente, en curso, denegado, terminado.	
Atributo	Tipo	Descripción
id_estado_de_visitante	serial	Este campo es una llave primaria y almacena el identificador que distingue a la tabla.
nombre	character varying	Este campo almacena el nombre de los estados del visitante.
Nombre_slug	character varying	Nombre único que diferencia los elementos.

Tabla 3.2 Descripción de la tabla: estado de visitante.

Nombre	área	
Descripción	Esta tabla almacena el nombre del área a la que pertenecerá un Directivo.	
Atributo	Tipo	Descripción
id_área	serial	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
Nombre	character varying	Este campo almacena el nombre del área a la que pertenece un Directivo.
Id_directivo	character varying	Este campo es una llave foránea y almacena el identificador de la tabla <i>directivo</i> .

Tabla 3.3 Descripción de la tabla: área.

Nombre	organismo	
Descripción	Esta tabla almacena la información referente a cada organismo.	
Atributo	Tipo	Descripción
id_organismo	character	Este campo es una llave primaria y almacena el

	varying	identificador que distingue a esta tabla.
Código	character varying	Este campo almacena el código que identifica a cada organismo o las siglas correspondientes al mismo.
Nombre	character varying	Este campo almacena el nombre del organismo al que pertenece el visitante.

Tabla 3.4 Descripción de la tabla: organismo.

Nombre	visita	
Descripción	Esta tabla almacena los datos de la visita.	
Atributo	Tipo	Descripción
id_visita	Integer	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
id_estado_de_visita	Varchar	Este campo es una llave foránea y almacena el identificador de la tabla <i>estado_de_visita</i> .
Fecha_inicio	Integer	Este campo almacena la fecha de inicio de la visita.
Fecha_fin	Integer	Este campo almacena la fecha de terminada la visita.

Tabla 3.5 Descripción de la tabla: visita

Nombre	categoría_de_solicitud	
Descripción	Esta tabla almacena las categorías que puede presentar una solicitud las cuales puedes ser: <i>común, imprevista, notificada, o prórroga</i> .	
Atributo	Tipo	Descripción
id_categoría_de_solicitud	Integer	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
nombre	character varying	Este campo almacena el nombre del tipo de solicitud realizada.
nombre_slug	character varying	Nombre único que diferencia los elementos.

Tabla 3.6 Descripción de la tabla: categoría_de_solicitud.

Nombre	tipo_de_solicitud.
Descripción	Esta tabla almacena la información referente a las categorías que posee una

solicitud que pueden ser: <i>diurna, temporal o al hotelito</i> .		
Atributo	Tipo	Descripción
id_tipo_solicitud	serial	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
Nombre	character varying	Este campo almacena el nombre correspondiente a la solicitud realizada.
Id_tipo_visitante	Integer	Este campo es una llave foránea y almacena el identificador de la tabla <i>tipo_visitante</i> .
nombre_slug	character varying	Nombre único que diferencia los elementos.

Tabla 3.6 Descripción de la tabla: tipo_solicitud.

Nombre	solicitud	
Descripción	Esta tabla almacena la información referente a las solicitudes.	
Atributo	Tipo	Descripción
id_solicitud	serial	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
Id_solicitante	varchar	Este campo es una llave foránea y almacena el identificador de la tabla <i>solicitante</i> .
Id_estado_de_solicitud	int	Este campo es una llave foránea y almacena el identificador de la tabla <i>id_estado_de_solicitud</i> .
Id_categoria_de_solicitud	Integer	Este campo es una llave foránea y almacena el identificador de la tabla <i>categoría_de_solicitud</i> .
Id_tipo_de_solicitud	Integer	Este campo es una llave foránea y almacena el identificador de la tabla <i>tipo_de_solicitud</i> .
Id_area	Integer	Este campo es una llave foránea y almacena el identificador de la tabla <i>área</i> .
Id_directivo	character varying	Este campo es una llave foránea y almacena el identificador de la tabla <i>directivo</i> .
Fecha_entrada	Timestamp without time zone	En este campo se almacena la fecha para la que se solicita la entrada del visitante.
Fecha_salida	Timestamp without time zone	En este campo se almacena la fecha para la que se solicita la salida del visitante.
Fecha_de_revision	Character	En este campo se almacena la fecha en que la

	varying	solicitud fue aprobada o denegada.
Motivo	Varchar	En este campo se almacenan los motivos por los cuales es realizada la solicitud
Motivo_de_negacion	Varchar	En este campo se almacenan los motivos por los cuales es negada la solicitud
Created_at	Timestamp without time zone	En este campo se almacena la fecha en que fue creada la solicitud
Updated_at	Timestamp without time zone	En este campo se almacena la fecha en que fue modificada la solicitud.

Tabla 3.7 Descripción de la tabla: solicitud.

Nombre	tipo_de_solicitud_tributo	
Descripción	Esta tabla almacena los atributos no comunes de la solicitud.	
Atributo	Tipo	Descripción
id_tipo_de_solicitud	Integer	Este campo almacena el identificador que distingue a esta tabla.
Id_tributo	integer	Este campo es una llave foránea y almacena el identificador de la tabla <i>atributo</i> .

Tabla 3.8 Descripción de la tabla: atributo_de_solicitud.

Nombre	visitante	
Descripción	Esta tabla almacena la información referente a los visitantes.	
Atributo	Tipo	Descripción
id_visitante	seriel	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
Nombre	Varchar	Este campo almacena el nombre del visitante.
P_apellido	Varchar	Este campo almacena el primer apellido del visitante.
S_apellido	Varchar	Este campo almacena el segundo apellido del visitante.
CI	bpchar	Este campo almacena en número de carnet de identidad del visitante.

Tabla 3.9 Descripción de la tabla: visitante.

Nombre	directivo	
Descripción	Esta tabla almacena la información referente a los directivos que acceden al sistema.	
Atributo	Tipo	Descripción
id_directivo	varchar	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.

Tabla 3.10 Descripción de la tabla: directivo.

Nombre	solicitante	
Descripción	Esta tabla almacena la información referente los solicitantes que acceden al sitio.	
Atributo	Tipo	Descripción
id_solicitante	varchar	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.

Tabla 3.11 Descripción de la tabla: solicitante

Nombre	Valor_atributo_visitante	
Descripción	Esta tabla almacena la información referente a los visitantes.	
Atributo	Tipo	Descripción
id_atributo	Integer	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
valor	Varchar	Este campo almacena el valor del atributo.
id_visitante	Integer	Este campo es una llave foránea y almacena el identificador de la tabla <i>tipo_visitante</i> .

Tabla 3.12 Descripción de la tabla: valor_atributo_visitante.

Nombre	Tipo_de_visitante	
Descripción	Esta tabla almacena la información referente a el tipo de visitante que pueden ser: <i>diurno</i> , <i>temporal</i> o <i>al hotelito</i> .	
Atributo	Tipo	Descripción
id_tipo_de_visitante	serial	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
Nombre	Varchar	Este campo almacena el nombre correspondiente al visitante según la categoría de su visita.

Nombre_slug	Varchar	Nombre único que diferencia los elementos.
-------------	---------	--

Tabla 3.13 Descripción de la tabla: tipo_de_visitante.

Nombre	solicitud_visitante	
Descripción	Esta tabla almacena la relación solicitud- visitante y el estado actual del visitante.	
Atributo	Tipo	Descripción
id_solicitud	Integer	Este campo es llave foránea y almacena el identificador de la tabla <i>solicitud</i> .
Id_visitante	Integer	Este campo es llave foránea y almacena el identificador de la tabla <i>visitante</i> .
Id_estado_de_visitante	integer	Este campo es llave foránea y almacena el identificador de la tabla <i>visitante</i> .

Tabla 3.14 Descripción de la tabla solicitud_visitante.

Nombre	valor_atributo_visitante	
Descripción	Esta tabla almacena la información referente al valor que toma el atributo del visitante.	
Atributo	Tipo	Descripción
id_atributo	Integer	Este campo es llave foránea y almacena el identificador del atributo.
Id_visitante	Integer	Este campo es llave foránea y almacena el identificador del visitante.
Valor	Varchar	Este campo almacena el valor que posee el atributo.

Tabla 3.15 Descripción de la tabla: valor_atributo_visitante.

Nombre	valor_atributo_de_solicitud	
Descripción	Esta tabla almacena la información referente al valor que toma el atributo de la solicitud.	
Atributo	Tipo	Descripción
id_solicitud	Integer	Este campo es llave foránea y almacena el identificador de la solicitud.
Id_atributo	Integer	Este campo es llave foránea y almacena el identificador del atributo

Valor	varchar	Este campo almacena el valor que toma el atributo.
-------	---------	--

Tabla 3.16 Descripción de la tabla: valor_atributo_de_solicitud.

Nombre	estado_de_solicitud	
Descripción	Esta tabla almacena la información referente a los posibles estados en que puede encontrarse una solicitud realizada que pueden ser: <i>pendiente, aprobada o denegada</i>	
Atributo	Tipo	Descripción
id_estado_de_solicitud	serial	Este campo es una llave primaria y almacena el identificador que distingue a esta tabla.
Nombre	Varchar	Este campo almacena el nombre del estado en que se encuentra la solicitud.
Nombre_slug	varchar	Nombre único que diferencia los elementos.

Tabla 3.17 Descripción de la tabla: estado_de_solicitud.

3.6 Tratamiento de errores

Para el tratamiento de errores se validan los datos que son introducidos por los usuarios al sistema. Se valida que el formato de los datos sea el esperado y que no se omita información de importancia para el procesamiento de solicitudes. En estos casos se informa al usuario mediante un mensaje que ha realizado alguna operación incorrecta. Cada usuario tendrá acceso restringido según su rol y se le permitirá solo ver las funcionalidades para las cuales obtenga permisos.

3.7 Seguridad

Para garantizar la seguridad de la información se crearon varios niveles, definidos como roles de usuario, que pueden ser: Trabajador UCI, Trabajador Interno, Técnico SP, Instructora, Especialista, Directivo, Servicios y Administrador. Este último es el encargado de los cambios y del buen funcionamiento del sistema, por tanto, poseerá control total del mismo. El resto de los usuarios solo tendrá acceso a la información determinada por el administrador del sistema según el rol que posea. Para esto, se utiliza el manejador de permisos integrado que tiene Symfony, el cual permite conocer que usuario intenta visitar dichas páginas y le será mostrado solo aquello a lo que debe tener acceso.

Además, el uso de LDAP (Lightweight Directory Access Protocol traducido al español como Protocolo Ligero de Acceso a Directorios) como protocolo de red permite el acceso a un servicio de directorio ordenado y distribuido ya que almacena la información de login (usuario y contraseña) y es utilizado para autenticarse.

3.8 Principios del diseño

El modelo de interacción entre el sistema y el usuario, es el diseño de la propia aplicación, por lo que requiere de significativa atención. El diseño debe tener en cuenta los siete Principios del Diseño Universal o Diseño para Todos: uso equiparable, uso flexible, simple e intuitivo, información perceptible, con tolerancia al error, que exija poco esfuerzo físico y por último, tamaño y espacio para el acceso y uso. Estos Principios generales del diseño se aplican en la arquitectura, la ingeniería y, por supuesto, las páginas y aplicaciones Web, entre otros campos de aplicación por lo que el cumplimiento de estos principios generales para el diseño permitirá la conquista de los objetivos pretendidos.

En el diseño de la aplicación se ha tenido en cuenta que:

- Las pantallas serán claras y directas, y en caso de repetirse alguna información, esta se situará en la misma posición.
- El sistema mostrará a cada usuario solo la información que le corresponde según el rol que posee en el sistema.
- Se utilizará un mismo color en todas las páginas y un mismo tipo y tamaño de letra con textos claros, el color predominante será el azul. El idioma utilizado es el español.

En todo momento en la parte superior del sistema se mostrará el usuario que está autenticado.

3.9 Conclusiones

Luego de concluir la fase de Análisis y Diseño, el aporte fundamental es el de haberse creado todos los artefactos necesarios para nutrir de detalles y especificaciones la futura implementación del sistema. Se obtuvo el modelo de análisis que sirvió de entrada fundamental a las actividades del diseño, de gran importancia pues en ella se definieron las clases con sus atributos y funcionalidades. Queda constituida ya la estructura del futuro sistema teniendo en cuenta todos los requisitos propuestos en la fase anterior. Se logra proponer una interfaz amigable y sencilla acorde con las exigencias del cliente, así como establecer medidas de seguridad que junto a un eficaz tratamiento de errores hacen fiable el futuro sistema.

CAPITULO 4 Implementación y Prueba

4.1 Introducción

El presente capítulo abarca la fase de implementación y pruebas que comienza con los resultados obtenidos del modelo de diseño y continúa con la implementación del sistema en términos de componentes para terminar en la aplicación de las pruebas de caja negra en pos de garantizar la calidad del sistema. Se obtendrán también los diagramas de componentes.

4.2 Implementación

Este flujo de trabajo enmarca el comienzo de la fase de Construcción. Esta fase tiene como propósito dejar listo un producto software en su versión operativa inicial (versión beta). A esta versión le incumbe tener la calidad requerida para su uso y cumplir con los requisitos de software determinados en el segundo capítulo.

El sistema que se implementa está compuesto por un proyecto nombrado *visitas*, éste contiene dos aplicaciones: frontend, con la cual interactúan los usuarios del sistema y backend para la administración.

En cada aplicación se crearon módulos que fueron básicamente los que dieron las funcionalidades al sistema. Los módulos del frontend son: *módulo solicitud*, *módulo visita*, *módulo área* y *módulo visitante* y en el backend: *area* y *directivo*. Una de las principales características de Symfony es la reutilización de código (no reinventar la rueda), este proyecto también aplica estas premisas y hace uso de módulos ya implementados por otros, evitando con esto el desgaste innecesario. Estos módulos son llamados plugins y son una de las principales potencialidades del framework, que cuenta con alrededor de 500. En este trabajo se utilizan tres de ellos:

- *sfFormExtraPlugin* : este plugin permitió añadir widgets a los formularios Symfony .(poner la foto de la fecha)
- *sfJqueryWidgetsPlugin*: Permite añadir widgets que no contenía el plugin anterior.
- *sfGuardPlugin*: Es este un modulo con un excelente mecanismo de seguridad implementado.

El proyecto se implementó en dos entornos: dev y prod. El primero es de desarrollo y el segundo de producción

En la implementación del sistema la adaptabilidad para las reglas del negocio fue fundamental, en capítulos anteriores se vieron los distintos tipos de solicitudes que se podían realizar, cada uno de estos presentaban datos comunes y específicos; el sistema tenía que ser capaz de lidiar con estas similitudes y diferencias y a la vez adaptarse a cambios que pudieran sufrir cada uno de éstos en cuanto a los datos necesarios con que se crean. En el modelo de entidad relacional de la base de datos se utilizó el patrón EAV para lidiar con esto, el otro reto que surgía era llevar esto a la interfaz que fuera algo transparente para el usuario.

Se ilustrará con un ejemplo lo explicado anteriormente:

Las solicitudes temporales necesitan del apartamento donde se visitará, sin embargo la diurna no presenta este dato, así mismo ésta última necesita del organismo de los visitantes, las solicitudes al hotelito, del sexo de los visitantes y las temporales, del parentesco. Esto conlleva a la doble utilización del EAV y por consiguiente que tanto la interfaz para las solicitudes como para los visitantes que están en cada una de ellas sea dinámica, o sea, que según el tipo, sepa que campos mostrar, así mismo, si en un tiempo se hace necesario eliminar o agregar un nuevo campo a algún tipo de solicitud o de visitante la adaptabilidad del sistema permitirá sin prácticamente tocar el código darle solución a este problema.

A continuación se explicará brevemente la dicha implementación.

Se definieron un conjunto de atributos posibles

atributos:

sexo:

widget: sfWidgetFormChoice

opciones:

choices: [F,M]

expanded: true

label: Sexo

validator: null

Lo que significa esta descripción es lo siguiente:

El atributo sexo será representado en un formulario por un `sfWidgetFormChoice` expandido y sus ítems serán F y M, la etiqueta será Sexo y no tendrá validador.

Cada atributo fue descrito de esta forma en el archivo `yaml`, `atributos.yml`, este archivo es procesado por una clase manejadora `AttManager` que se encarga de crear el control visual y los validadores según la configuración entrada.

Otra funcionalidad importante que se logró implementar en el sistema fue la no persistencia física de visitantes mientras se crea una solicitud, cosa que no es muy común, y que requiere de un esfuerzo extra. A continuación se procederá brevemente a explicar lo dicho.

Cuando un solicitante está realizando una solicitud, debe ir insertando visitantes (los visitantes que desea que ingresen al centro), pero estos visitantes no deben ser insertados en la base de datos, primero porque la solicitud puede no estar ni siquiera creada y segundo, porque no sería factible estar haciendo operaciones en la base de datos que posiblemente no perduren, pues el solicitante mientras confecciona la solicitud puede eliminar, actualizar o insertar.

Este problema es muy similar al problema del “carrito de compras”, y fue en lo que se basó la implementación, la diferencia radicaba en que los visitantes debían mostrarse en un listado; donde se editaran y eliminaran, por lo que se debía tener identificado cada uno de ellos. Lo que se hizo para solucionar esto fue crear un token para cada visitante virtual, que lo identificara durante todo el proceso de creación y edición de solicitudes. Para ello, como se recomienda, se trabajó con las sesiones, aspecto que `Symfony` brinda de forma excelente, pero sin embargo si no se trabaja con disciplina puede acarrear diversos problemas, luego la solución consistió en apoyarse en el mecanismo de filtros del framework, éste consiste en definir clases, que vendrían siendo estados, por los que debe pasar una petición antes de ejecutarse del todo. De esta forma se creó un filtro `sessionFilter`, que hereda de `sfFilter`, clase que tiene todas las funcionalidades de filtro del framework, y se definieron un conjunto de reglas que definían según la petición que variables de sesión (por ejemplo la lista de visitantes) debía ser destruida o mantenida.

4.3 Diagramas de Componentes

Un diagrama de componentes se representa como un grafo de componentes unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten. Cada componente representa una parte modular del sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos.

Estos diagramas son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos. Dicho diagrama contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema y la distribución de las partes del sistema en ellos.

A continuación se muestra el diagrama de componentes de los casos de uso significativos del sistema. El resto de los diagramas podrán ser vistos en el anexo #8.

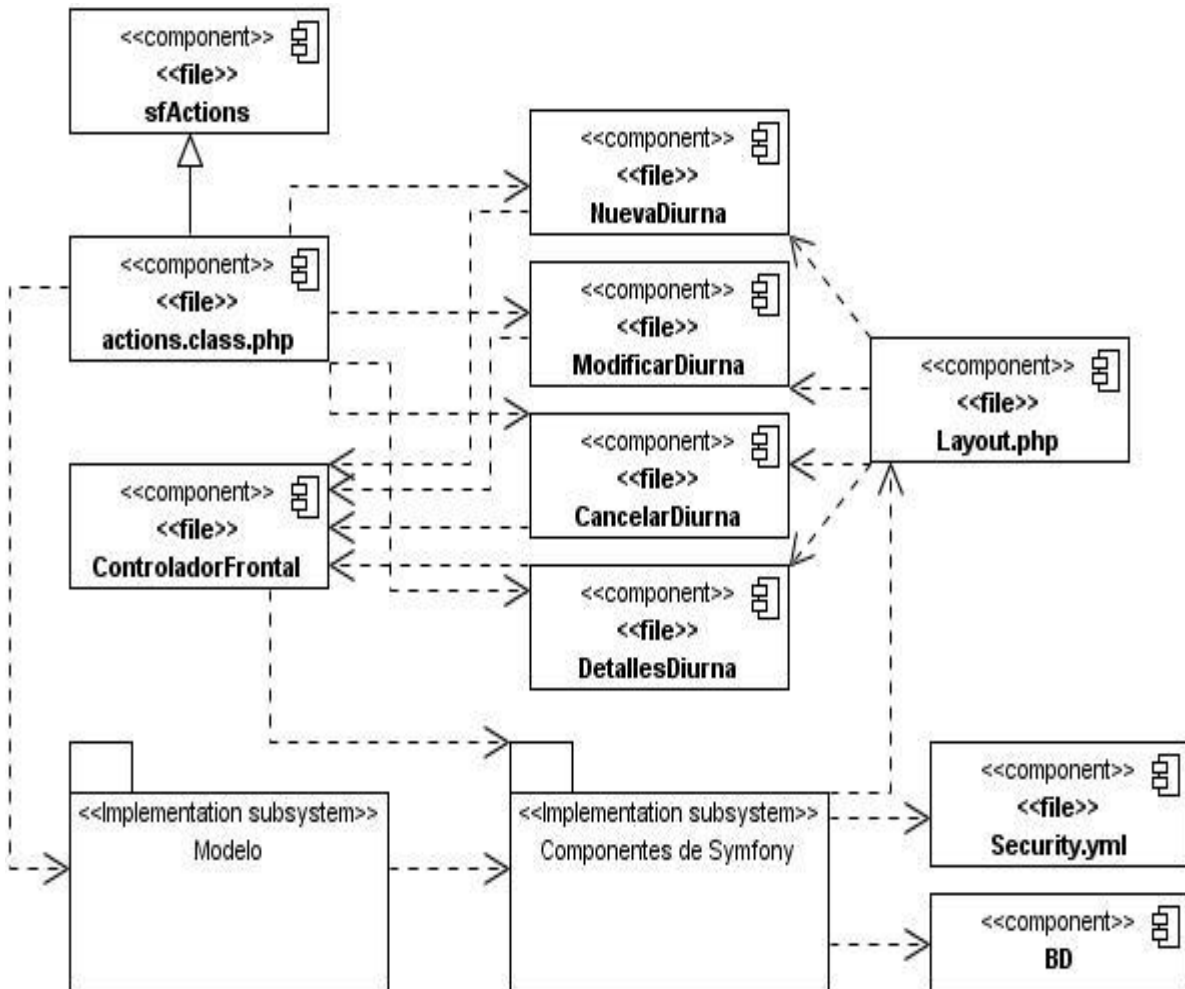


Figura 26 Diagrama de componentes del CU: Gestionar visita diurna.

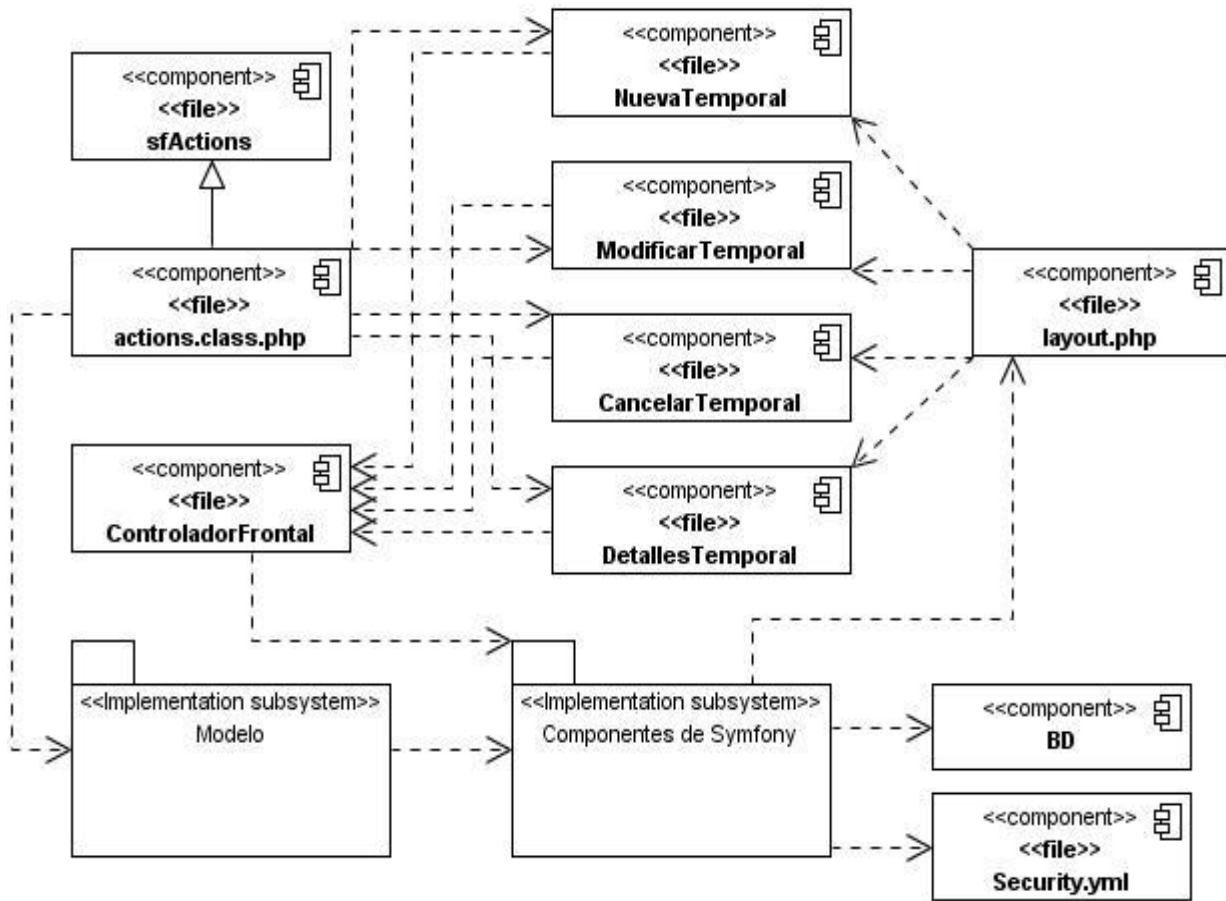


Figura 27 Diagrama de componentes del CU: Gestionar visita temporal.

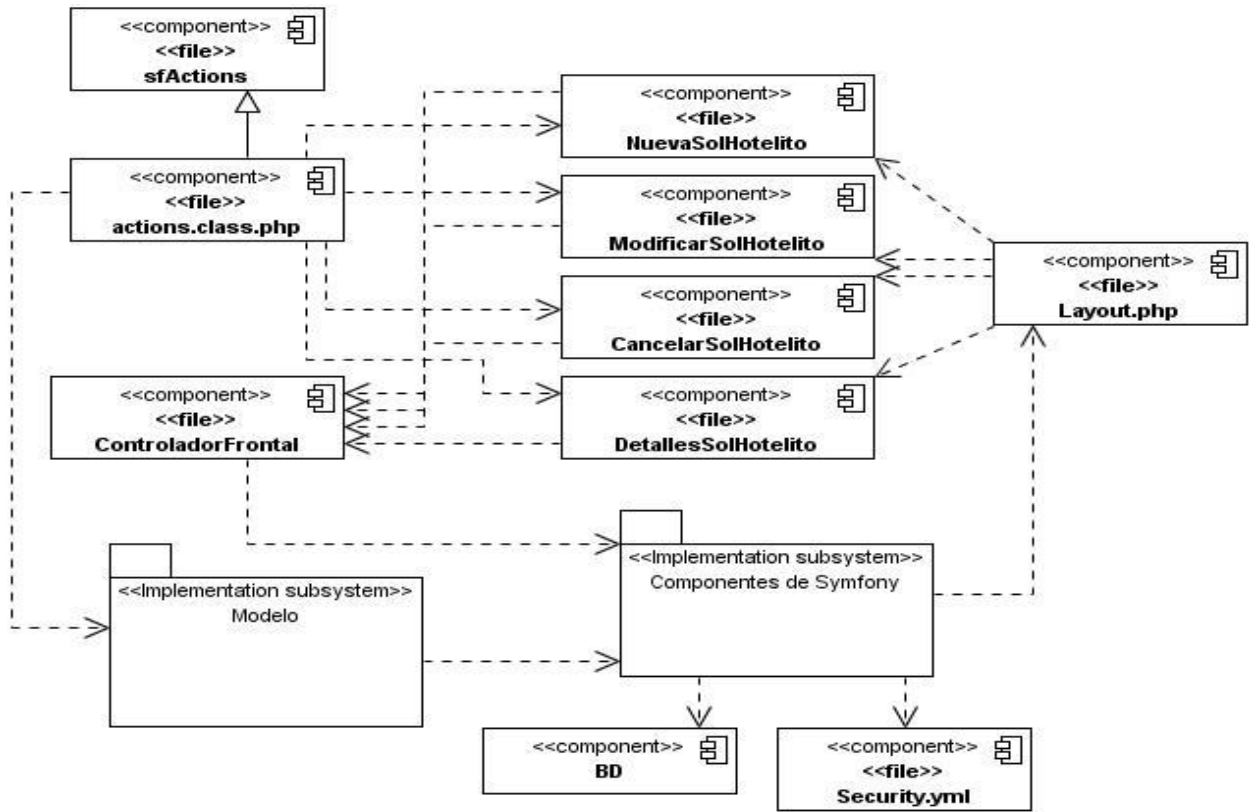


Figura 28 Diagrama de componentes del CU: Gestionar visita al hotelito.

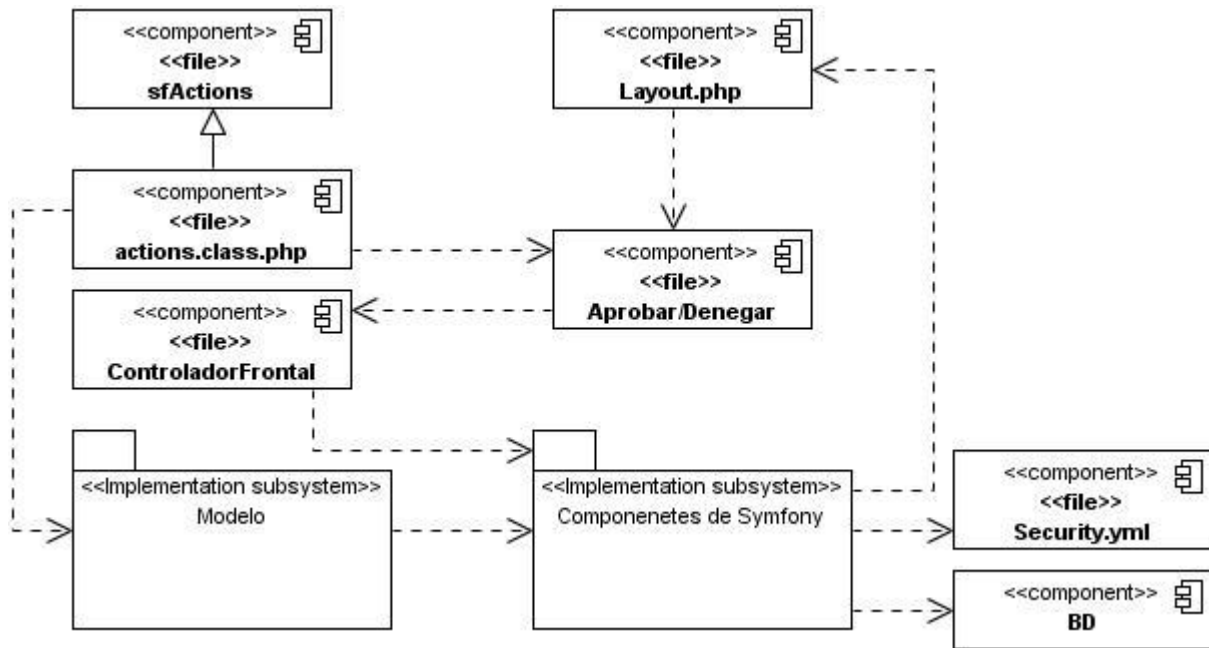


Figura 29 Diagrama de componentes del CU: Aprobar/denegar solicitudes.

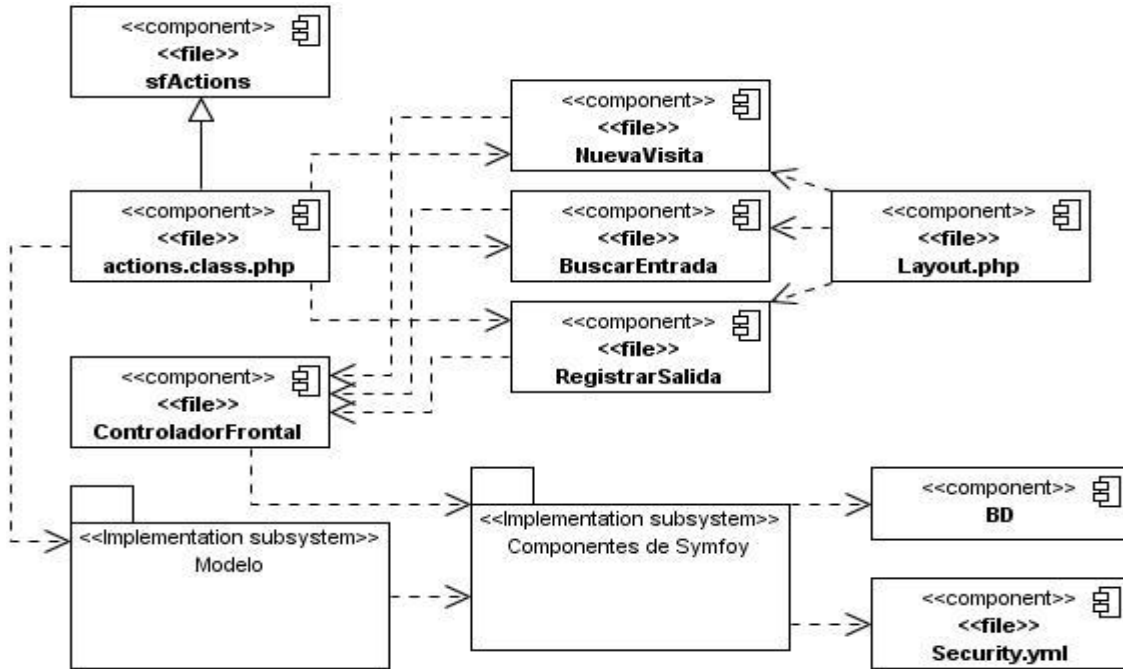


Figura 30 Diagrama de componentes del CU: Registrar entrada/salida de visitantes.

4.4 Diagrama de Despliegue

Es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Gráficamente, un diagrama de despliegue es una colección de nodos y arcos. [13]

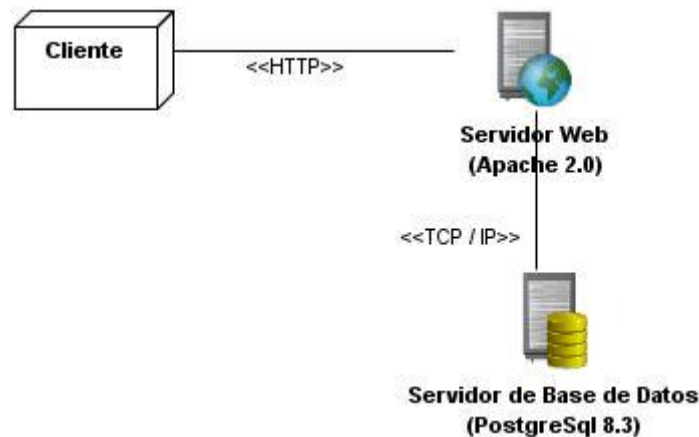


Figura 31 Diagrama de despliegue del sistema.

4.5 Pruebas

En el desarrollo de un software específico, el proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan a la calidad de un producto bien desarrollado. Las pruebas son de gran importancia en la garantía del software, una selección cuidadosa de los datos de prueba puede ofrecer mucha confianza en cuanto al desempeño que posee el programa. Esto, asociado a un determinado mecanismo de comprobación de errores, puede producir software más confiable. Los objetivos principales de la realización de una prueba son tener un buen caso de prueba que sirva para descubrir un error no descubierto antes.

4.5.1 Métodos de Prueba

Los Métodos de Prueba de Software tienen el objetivo de diseñar pruebas que descubran diferentes tipos de errores con menor tiempo y esfuerzo.

Existen dos métodos de prueba:

Pruebas de caja negra: El sistema de pruebas de caja negra no considera la codificación dentro de sus parámetros a evaluar, es decir, no están basadas en el conocimiento del diseño interno del programa. Estas pruebas se enfocan en los requerimientos establecidos y en la funcionalidad del sistema.

Pruebas de caja blanca: Al contrario de las de caja negra, estas se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas completan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc.
[18]

4.5.2 Prueba del Sistema

El método de prueba seleccionado fue el método de caja negra y dentro del mismo se desarrollaron las pruebas de sistema. Estas pruebas son realizadas una vez concluido el software y luego que todos los componentes de software y hardware han sido integrados. Las pruebas de sistema principalmente se centran en verificar la interacción de los actores con el sistema.

Los datos de prueba se escogerán atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa se ejecute correctamente. Criterios mínimos que guiarán al escoger los datos de prueba:

Valores fáciles: El programa se depurará con datos fáciles de comprobar.

Valores típicos realistas: Se ensayará un programa con datos seleccionados para que representen cómo se aplicará. Estos datos han de ser suficientemente sencillos, de modo que los resultados sean verificables en forma manual, valores futuros candidatos a ser usados en la aplicación.

Valores extremos: Muchos programas cometen errores en los límites de sus rangos de aplicaciones.

Valores ilegales: Se entrarán datos cuyos valores no corresponden con los requeridos por el sistema, su reacción inmediata habrá de ser por lo menos un mensaje de error adecuado para el usuario.

A continuación se muestran los principales casos de prueba:

- CP 1: Nombre del caso de uso: Gestionar Solicitud Diurna

Sección: Nueva Solicitud Diurna

Entrada	Resultados	Condiciones
El Trabajador UCI introduce los datos correctamente y presiona el botón Aceptar.	El sistema adiciona una nueva solicitud a la base de datos.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe introducir los datos correctamente.
El Trabajador UCI deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe dejar al menos un campo vacío.
El Trabajador UCI introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe introducir al sistema datos incorrectos.

Tabla 4.1 CP1. CU Gestionar Solicitud Diurna. Sección: Nueva Solicitud Diurna

Sección: Modificar Solicitud Diurna

Entrada	Resultados	Condiciones
El Trabajador UCI introduce los datos correctamente y presiona el botón Aceptar.	El sistema modifica la solicitud y guarda los cambios en la base de datos.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe introducir los datos correctamente.
El Trabajador UCI deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe dejar al menos un campo vacío.
El Trabajador UCI introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe introducir al sistema datos incorrectos.

Tabla 4.2 CP1. CU Gestionar Solicitud Diurna. Sección: Modificar Solicitud Diurna

Sección: Cancelar Solicitud Diurna

Entrada	Resultados	Condiciones
El Trabajador UCI selecciona el link Cancelar de una solicitud realizada anteriormente	El sistema elimina esa solicitud del listado de solicitudes enviadas y actualiza la base de datos.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe seleccionar el link Cancelar.

Tabla 4.3 CP1. CU Gestionar Solicitud Diurna. Sección: Cancelar Solicitud Diurna

Sección: Detalles de Solicitud Diurna

Entrada	Resultados	Condiciones
El Trabajador UCI selecciona el link Detalles de una solicitud realizada anteriormente	El sistema muestra una vista con todos los datos de la solicitud seleccionada.	El usuario debe estar autenticado en el sistema con el rol Trabajador UCI y debe seleccionar el link Detalles.

Tabla 4.4 CP1. CU Gestionar Solicitud Diurna. Sección: Detalles Solicitud Diurna

➤ CP 2: Nombre del caso de uso: Gestionar Solicitud Temporal

Sección: Nueva Solicitud Temporal

Entrada	Resultados	Condiciones
El Trabajador Interno introduce los datos correctamente y presiona el botón Aceptar.	El sistema adiciona una nueva solicitud a la base de datos.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe introducir los datos correctamente.
El Trabajador Interno deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe dejar al menos un campo vacío.
El Trabajador Interno introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe introducir al sistema datos incorrectos.

Tabla 4.5 CP2. CU Gestionar Solicitud Temporal. Sección: Nueva Solicitud Temporal.

Sección: Modificar Solicitud Temporal

Entrada	Resultados	Condiciones
El Trabajador Interno introduce los datos correctamente y presiona el botón Aceptar.	El sistema modifica la solicitud y guarda los cambios en la base de datos.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe introducir los datos correctamente.
El Trabajador Interno deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe dejar al menos un campo vacío.

El Trabajador Interno introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe introducir al sistema datos incorrectos.
---	---	---

Tabla 4.6 CP2. CU Gestionar Solicitud Temporal. Sección: Modificar Solicitud Temporal.

Sección: Cancelar Solicitud Temporal.

Entrada	Resultados	Condiciones
El Trabajador Interno selecciona el link Cancelar de una solicitud realizada anteriormente	El sistema elimina esa solicitud del listado de solicitudes y actualiza la base de datos.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe seleccionar el link Cancelar.

Tabla 4.7 CP2. CU Gestionar Solicitud Temporal. Sección: Cancelar Solicitud Temporal.

Sección: Detalles de Solicitud Temporal.

Entrada	Resultados	Condiciones
El Trabajador Interno selecciona el link Detalles de una solicitud realizada anteriormente	El sistema muestra una vista con todos los datos de la solicitud seleccionada.	El usuario debe estar autenticado en el sistema con el rol Trabajador Interno y debe seleccionar el link Detalles.

Tabla 4.8 CP2. CU Gestionar Solicitud Temporal. Sección: Detalles Solicitud Temporal.

➤ CP 3: Nombre del caso de uso: Gestionar Solicitud Hotelito

Sección: Nueva Solicitud Hotelito

Entrada	Resultados	Condiciones
El Directivo introduce los datos correctamente y presiona el botón Aceptar.	El sistema adiciona una nueva solicitud a la base de datos.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe introducir los datos

		correctamente.
El Directivo deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe dejar al menos un campo vacío.
El Directivo introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe introducir al sistema datos incorrectos.

Tabla 4.9 CP3. CU Gestionar Solicitud Hotelito. Sección: Nueva Solicitud Hotelito.

Sección: Modificar Solicitud Hotelito

Entrada	Resultados	Condiciones
El Directivo introduce los datos correctamente y presiona el botón Aceptar.	El sistema modifica la solicitud y guarda los cambios en la base de datos.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe introducir los datos correctamente.
El Directivo deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe dejar al menos un campo vacío.
El Directivo introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe introducir al sistema datos incorrectos.

Tabla 4.10 CP3. CU Gestionar Solicitud Hotelito. Sección: Modificar Solicitud Hotelito.

Sección: Cancelar Solicitud Hotelito

Entrada	Resultados	Condiciones
El Directivo selecciona el link Cancelar de una solicitud	El sistema elimina esa solicitud del listado de solicitudes y	El usuario debe estar autenticado en el sistema con

realizada anteriormente	actualiza la base de datos.	el rol Directivo y debe seleccionar el link Cancelar.
-------------------------	-----------------------------	---

Tabla 4.11 CP3. CU Gestionar Solicitud Hotelito. Sección: Cancelar Solicitud Hotelito.

Sección: Detalles de Solicitud Hotelito

Entrada	Resultados	Condiciones
El Directivo selecciona el link Detalles de una solicitud realizada anteriormente	El sistema muestra una vista con los todos los datos de la solicitud seleccionada.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe seleccionar el link Detalles.

Tabla 4.12 CP3. CU Gestionar Solicitud Hotelito. Sección: Detalles Solicitud Hotelito.

➤ CP 4: Nombre del caso de uso: Aprobar/Denegar Solicitudes

Entrada	Resultados	Condiciones
El Directivo selecciona la solicitud a aprobar y presiona el botón Aceptar.	El sistema elimina esa solicitud del listado de las solicitudes recibidas pendientes de autorización.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe tener al menos una solicitud pendientes de aprobación en su área.
El Directivo selecciona una solicitud e introduce los motivos de su negación y presiona el botón denegar	El sistema elimina la solicitud del listado de solicitudes recibidas.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe tener solicitudes pendientes de aprobación en su área y realiza correctamente todo el procedimiento para denegar una solicitud.
El Directivo no introduce los motivos de su negación y presiona el botón Denegar	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe tener solicitudes pendientes de aprobación en su área y deja el campo "Motivos de negación" vacío.

Tabla 4.13 CP4. CU Aprobar/Denegar Solicitudes.

➤ CP 5 Registrar Entrada/Salida

Sección Nueva Visita

Entrada	Resultados	Condiciones
El Directivo introduce los datos correctamente y presiona el botón Aceptar.	El sistema adiciona una nueva visita a la base de datos.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe introducir los datos correctamente.
El Directivo deja al menos un campo vacío y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe dejar al menos un campo vacío.
El Directivo introduce datos no válidos y presiona el botón Aceptar.	El sistema muestra un mensaje de error.	El usuario debe estar autenticado en el sistema con el rol Directivo y debe introducir al sistema datos incorrectos.

Tabla 4.14 CP5. CU Registrar Entrada/Salida. Sección Nueva Visita

Sección Registrar Entrada/Salida de Visita

Entrada	Resultados	Condiciones
El Técnico SP introduce el número de carnet de identidad del visitante.	El sistema muestra los resultados de la búsqueda con los datos encontrados de la visita, al mismo tiempo que el Técnico SP introduce el número de carnet de identidad.	El usuario debe estar autenticado en el sistema con el rol Técnico SP y debe existir la visita buscada.
El Técnico SP introduce datos no válidos.	El sistema notifica el error.	El usuario debe estar autenticado en el sistema con el rol Técnico SP e introduce datos que no corresponden con los solicitados.
El Técnico SP no llena el campo "Observaciones" y presiona el botón "Registrar Salida"	El sistema notifica el error.	El usuario debe estar autenticado en el sistema con el rol Técnico SP y deja vacío el campo "Observaciones".

Tabla 4.15 CP5. CU Registrar Entrada/Salida. Sección Registrar Entrada/Salida de Visita

4.6 Conclusiones

Con la conclusión de este capítulo finaliza la implementación del Módulo de Visitas para el Sistema de Control de Acceso junto con su fase de pruebas de aseguramiento de la calidad del software. Fueron implementadas las clases obtenidas en el diseño. Se obtuvieron los diagramas de componentes que describían los módulos a construir y el de despliegue que muestra su organización. Fueron realizadas las pruebas al sistema con resultados satisfactorios que demuestran que la aplicación cuenta con las características y funcionalidades para las que fue concebido.

CONCLUSIONES GENERALES

Tras desarrollar y analizar los resultados obtenidos con la elaboración del presente trabajo se puede arribar a las siguientes conclusiones:

- Tras el desarrollo del presente trabajo se logra mejorar la gestión de solicitudes y aprobaciones de visitas a la Universidad, transformándose ahora en un proceso ágil, seguro y sencillo, de gran beneficio para los usuarios.
- Se logra mantener el control de las visitas que acceden al centro, descartando la posibilidad de pérdida de información y falsificación de documentos.
- Se logra alta fiabilidad a la hora de permitir el acceso de visitas, solo está registrado el personal que haya sido autorizado. Eliminando con esto los incómodos trámites telefónicos y personales.
- El uso de las herramientas seleccionadas fue factible y el desarrollo de la aplicación quedó ampliamente documentado mediante los artefactos generados por RUP para posibles adaptaciones o mejoras futuras.

De manera general se concluye que los objetivos planteados en la fase inicial fueron cumplidos favorablemente.

RECOMENDACIONES

Teniendo en cuenta los resultados obtenidos con el sistema implementado se recomienda:

- La puesta en funcionamiento del sistema para beneficio de los distintos usuarios.
- Integrar la aplicación al conjunto de módulos correspondientes al desarrollo de la Intranet 2 desplegado por el proyecto ERP Universitario.
- Desarrollar nuevas funcionalidades al sistema que permitan la evolución del mismo a la par de los procesos del negocio.
- Continuar el desarrollo del sistema de manera que pueda ser utilizado no sólo en la universidad, sino en cualquier empresa que demande la gestión de solicitudes y autorizaciones de visita.
- La utilización del presente trabajo como futuro material de consulta para la preparación de nuevas generaciones, sugiriendo también la bibliografía documentada.

REFERENCIAS BIBLIOGRAFICAS

1. avances.com.pe. [En línea] 2009. [Citado el: 10 de 01 de 2009.] <http://www.avances.com.pe/productos.html>).
2. ti2000.es. [En línea] 2006. [Citado el: 20 de 02 de 2009.] <http://www.ti2000.es/index.php?opcion=qaccess>.
3. intelektron.com. [En línea] 2003. [Citado el: 5 de 05 de 2009.] <http://www.intelektron.com/sitioacc/visitas/visitas.htm>.
4. sage.com.ar. [En línea] 2003. [Citado el: 15 de 12 de 2008.] <http://www.sage.com.ar/pages/ocean.html>.
5. security.honeywell.comesaccessproductosvms63310.html. [En línea] <http://www.security.honeywell.comesaccessproductosvms63310.html>.
6. zonafrancavigo.es. [En línea] [Citado el: 10 de 05 de 2009.] <http://www.zonafrancavigo.es/sc/sgc/contacteNosotros/solicitudVisita/solicitudVisita.html>.
3. caribscuba.com. [En línea] 2009. [Citado el: 20 de 02 de 2009.] <http://www.caribscuba.com>.
8. **Kon, Marcos David.** monografias.com. [En línea] 2009. [Citado el: 20 de 02 de 2009.] <http://www.monografias.com/trabajos12/elsoflib/elsoflib.shtml#definic>.
9. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*.
10. *Fase de Inicio. Modelo del Negocio.* Ciudad de la Habana : s.n., 2008.
11. *Fase de Inicio. Flujo de Trabajo de Requerimientos.* Ciudad de la Habana : s.n., 2009.
12. **Peralta, Mario.** *Estimación del Esfuerzo Basada en Casos de Uso.* s.l. : Reportes Técnicos en Ingeniería de Software, 2004. págs. 1-16. 1668-3133.
13. *Fase de Elaboración. Flujo de Trabajo de Análisis y Diseño.* Ciudad de la Habana : s.n., 2009.
14. adictosaltrabajo.com. [En línea] 22 de 12 de 2003. [Citado el: 23 de 02 de 2008.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=grasp..>
15. **Fabien Potencier, François Zaninotto.** *Symfony la guía definitiva.* 2008.
16. **Jacobson, Ivar; Booch, Grady; Rumbaugh, James.** *El Proceso Unificado de Software.*

17. **F. Días, J.** galeon.com. [En línea] [Citado el: 12 de 03 de 2009.]
http://www.galeon.com/neoprogramadores/met_test.htm.

BIBLIOGRAFIA

Apache.org [En línea] = Vision general de las nuevas funcionalidades de Apache 2.0. - The Apache software foundation, 2009. - 18 de 04 de 2009. - apache.org/docs/2.0/es/new_features_2_0.

Ciberaula.com [En línea] = Una Introducción a APACHE. - 2006. - 02 de 03 de 2009. - linux.ciberaula.com/articulo/linux_apache_intro/.

Di Biase Jose Luis y Toledo Fernando Desarrollo d eaplicaciones con Symfony [Conferencia]. - 2008.

Di Biase José luis y Toledo Fernando Desarrollo de Aplicaciones con Symfony [Libro].

Di Biase Jose Luis, Sanchez Hector Daniel y Toledo Fernando Primer talles de desarrollo con software libre Posadas - Misiones [Conferencia]. - 2009.

Di Biase Jose Luis, Sanchez Hector y Toledo Fernando Desarrollo de Aplicaciones con Symfony [Conferencia] // Taller sobre Symfony. - 2008.

El portal de la cultura matancera [En línea] = Estado actual de Eclipse. - Direccion Provincial de Cultura., 16 de 05 de 2008. - 16 de 03 de 2009. - www.atenas.cult.cu/?q=informatica.

gnu.org [En línea]. - Free Software Foundation, Inc., .

gnu.org [En línea] = The Free Software Definition. - Free Software Foundation, Inc, 26 de 04 de 2009. - 20 de 04 de 2009. - www.gnu.org/philosophy/free-sw.

koopmaschap Stefan Which plugind are out there? [Conferencia] // Symfony Camp. - [s.l.] : Dutch Open Project, 08- 09-2003.

Lellelid Hans Ping: Building in PHP [Conferencia]. - 2003-11-06.

Lellelid Hans Propel Your PHP Aplications [Conferencia] // International PHP Conference. - 2003-11-06.

Potencier Fabien Beyond Symfony 1.2 [Conferencia]. - [s.l.] : SensioLabs.

Potencier Fabien El tutorial Jobbet [Libro]. - 2009.

Potencier Fabien sensiolabs.com [En línea] // Symfony-proyect.com. - 2003. - 15 de 04 de 2009.

Potencier Fabien Symfony [Libro]. - [s.l.] : SensioLabs, 2008.

Potencier Fabien Symfony (An Open-Source Framework for professionals) [Conferencia].

Potencier Fabien Symfony. Simplify your professional web development whit PHP [Conferencia] // international PHP 2003 conference. - 2003.

Potencier Fabien The Synfony Platform [Conferencia] // Conference PHP Québec.

Symfony.es [En línea]. - 13 de mayo de 2009. - 18 de 01 de 2009. - www.symfony.es.

Van der Ree Leon Usable and Adaptable Bussiness Software [Libro]. - Netherlands : [s.n.], 2008.

Whittle Dustin Diving into Symfony [Conferencia]. - [s.l.] : Social Seach Group, 2009.

Whittle Dustin Lessons learned at Yahoo [Conferencia] // Symfony "Camp".

Whittle Dustin Symfony [Conferencia] // Web 2.0 Expo.

GLOSARIO DE TÉRMINOS Y SIGLAS

Personal ajeno: Toda aquella persona que no pertenece a la plantilla de trabajadores de la UCI.

Licencia BSD: Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software libre y a la vez permite el uso del código fuente en software no libre.

GPL: Sus siglas significan General Public License. Licencia diseñada para asegurar las libertades de distribución de copias de Software Libre, asegurar que se obtendrá el código fuente del programa, asegurar que se puede modificar el programa o modificar algunas de sus piezas para un nuevo programa.

Herramienta CASE: Aplicación informática destinada a aumentar la productividad en el desarrollo de software.

HTML: Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Manifiesto Ágil: Manifiesto firmado en el año 2001 que expone al descubierto mejores métodos para desarrollar software.

JVM: Sus siglas significan Máquina virtual de Java. JVM es únicamente un elemento del software de Java, específicamente utilizado para la interacción en la Web, que se incluye en la descarga del software de Java.

.Net Framework: Es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones. Mediante esta herramienta se ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.

J2EE: Sus siglas significan Java 2 Enterprise Edition. Define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por Sun Microsystems. J2EE simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando muchos de

las funciones de la aplicación de forma automática, sin necesidad de una programación compleja.

GoF: Gang of four, grupo compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, publicaron el libro Design Patterns en el que se recogían 23 patrones diseño comunes.

SQL: Structured Query Language

ANEXOS

Anexo #1 Diagrama de actividades

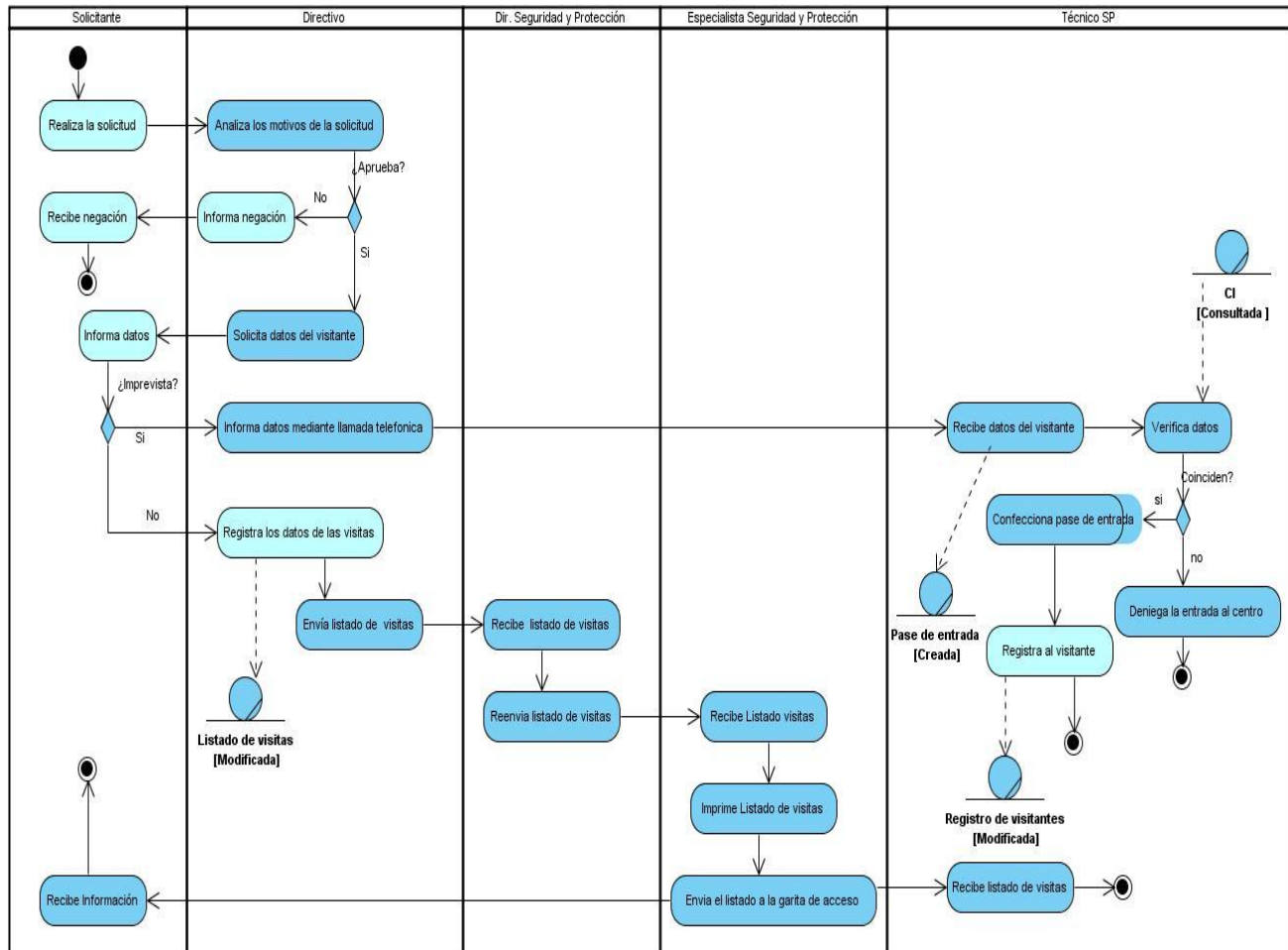


Diagrama de actividades del caso de uso: Solicitar visita diurna.

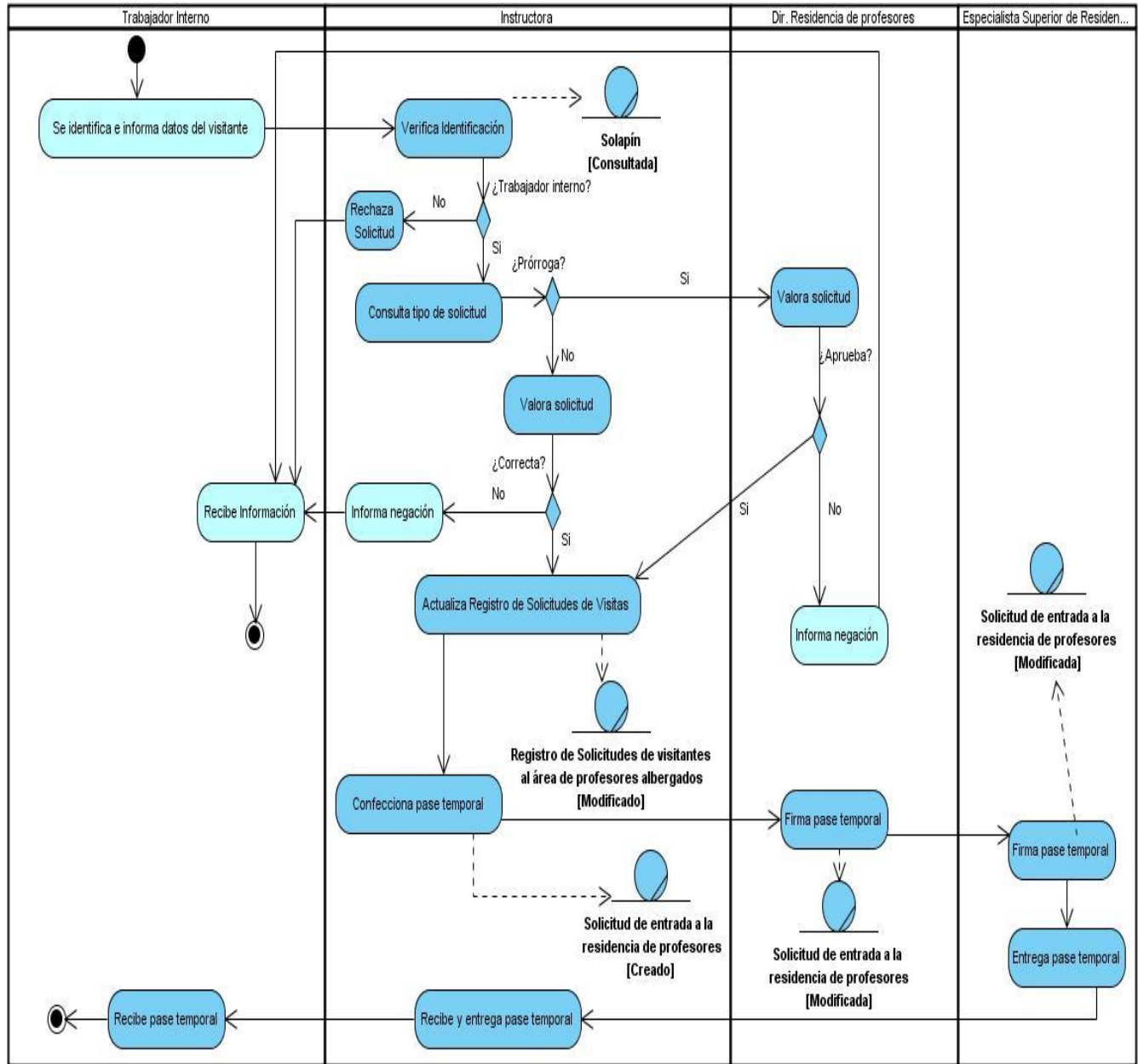


Diagrama de actividades del caso de uso: Solicitar visita temporal.

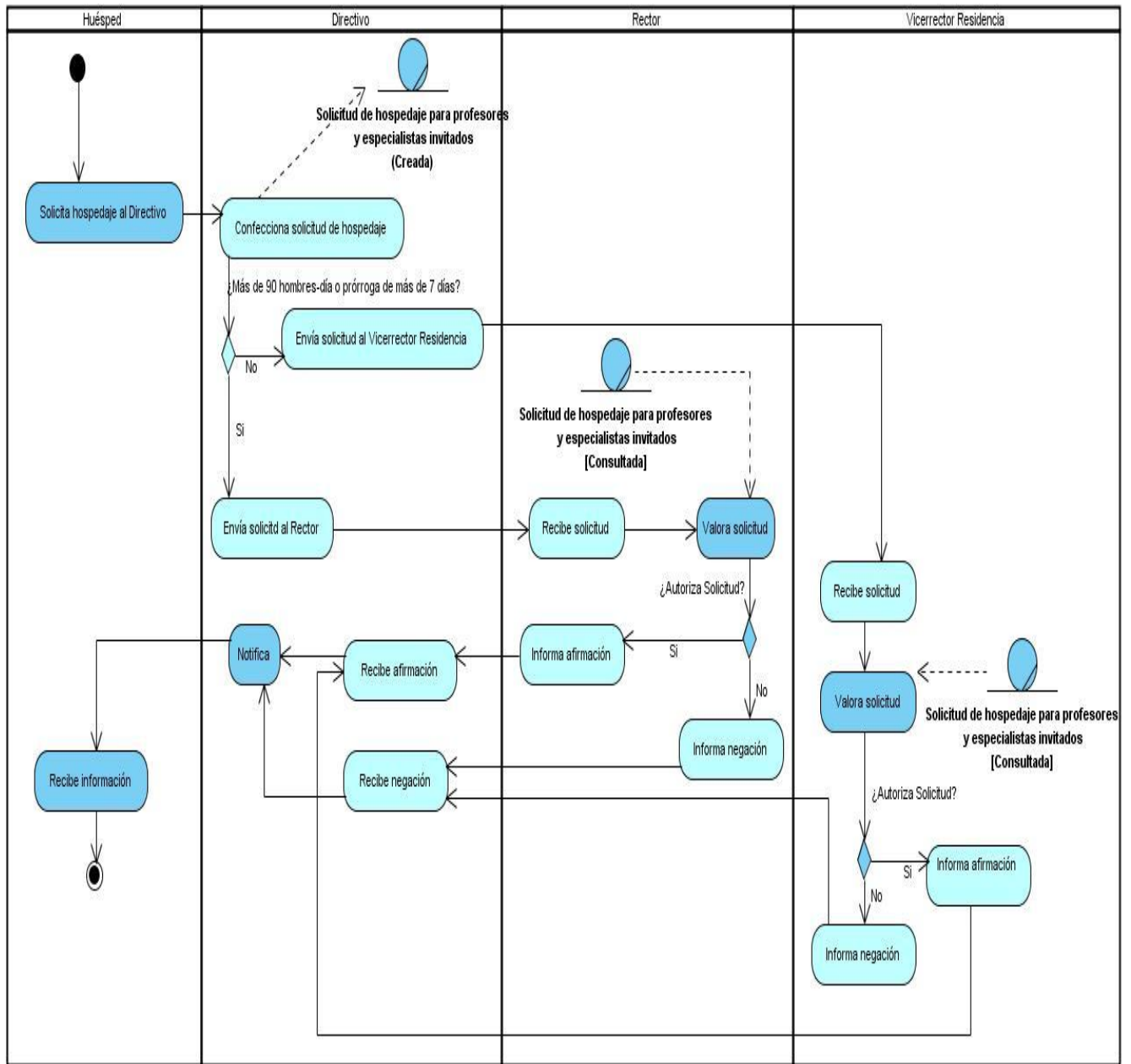
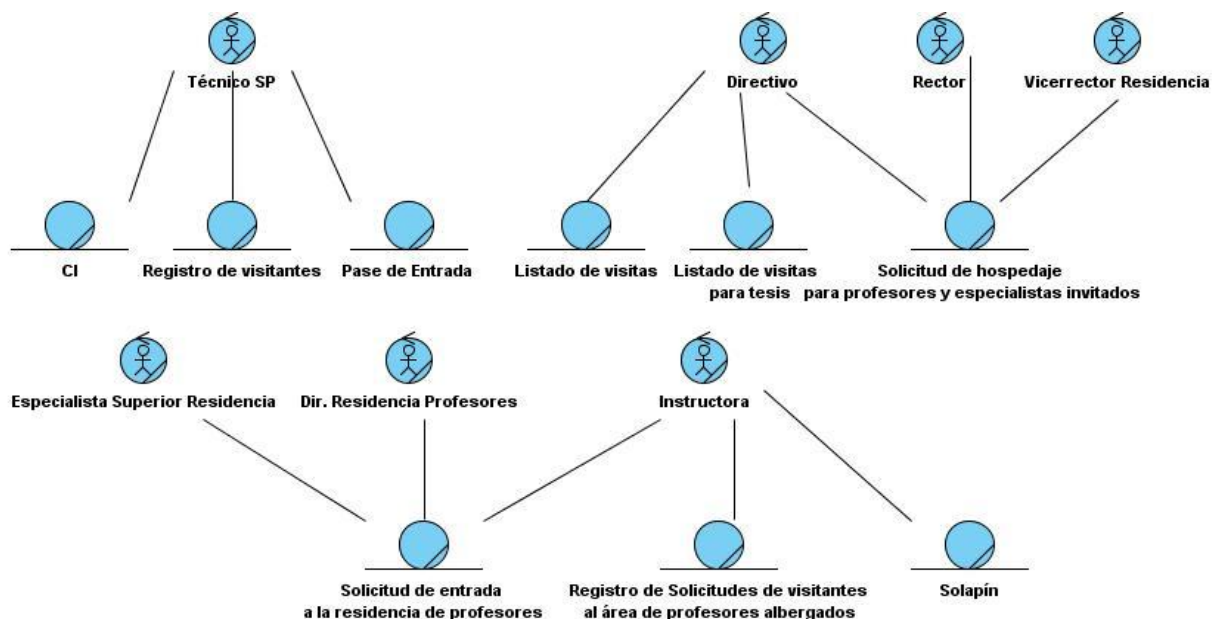


Diagrama de actividades del caso de uso: Solicitar visita al hotelito.

Anexo #2 Modelo de objetos del negocio



Anexo #3 Descripción de los casos de uso del sistema.

Caso de Uso.	Gestionar solicitud diurna.
Actor	Trabajador UCI.
Resumen.	
El caso de uso comienza cuando el trabajador UCI selecciona alguna de las opciones correspondientes al caso de uso. El usuario podrá realizar una nueva solicitud diurna y a partir del listado de las <i>Solicitudes Enviadas</i> que le será mostrado, podrá modificar o cancelar una solicitud del mismo, pudiendo ver también los detalles de cada solicitud en el listado de <i>Ultimas Enviadas</i> .	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R2, R2.1,R2.1.1, R2.1.1.1, R2.1.2, R2.1.2.1, R2.3, R2.2, R2.2.1, R2.2.1.1, R2.2.2, R2.2.2.1, R2.3, R2.3.1, R2.3.1.1, R 2.3.2, R2.4, R2.4.1
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. El trabajador UCI accede a la página principal.	1.1 El sistema le muestra el listado de las <i>Solicitudes Enviadas</i> que son las que están pendientes de aprobación y la posibilidad de modificar o cancelar cualquiera de estas solicitudes, así como de realizar una nueva solicitud.

<p>2 El Trabajador UCI selecciona una de las opciones.</p> <p>Si selecciona <i>Solicitud - Diurna (A)</i> (ver sección Nueva Solicitud Diurna y Plantilla 1).</p> <p>Si selecciona <i>Modificar (B)</i> (ver sección Modificar Solicitud Diurna y Plantilla 2 y 3).</p> <p>Si selecciona <i>Cancelar (C)</i> (ver sección Cancelar Solicitud Diurna).</p> <p>Si selecciona <i>Detalles (D)</i> (ver sección Detalles Solicitud Diurna y Plantilla 4).</p>	
Sección Nueva Solicitud Diurna	
	<p>2.1 Se le muestran todos los campos que debe llenar para la solicitud (fecha de entrada, organismo y motivos de la solicitud) así como la posibilidad de insertar todos los visitantes que desee (E) (nombre del visitante y carnet de identidad). Permite insertar visitantes que hayan sido registrados con anterioridad (F)</p>
<p>2.2 El trabajador UCI introduce los datos y los envía.</p>	<p>2.3 El sistema verifica que estén todos los campos llenos.</p> <p>2.4 El sistema valida los datos y actualiza.</p>
Sección Modificar Solicitud Diurna	
	<p>3.1 El sistema muestra todos los campos con la información de la solicitud escogida y la posibilidad de modificarlos. (G)</p>
<p>3.2 El trabajador UCI modifica los datos que desea y los envía.</p>	<p>3.3 El sistema verifica que estén todos los campos llenos.</p> <p>3.4 El sistema valida los datos y actualiza.</p>
Sección Cancelar Solicitud Diurna	
	<p>4.1 Muestra el mensaje: "Realmente desea cancelar la solicitud".</p>
<p>4.2 El trabajador UCI acepta.</p>	<p>4.3 El sistema actualiza las solicitudes.</p>
Sección Detalles Solicitud Diurna	

	5.1 Se le muestran toda la información de la solicitud.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema.
1.2 a) Seleccionar <i>Eliminar</i> si desea eliminar un visitante de la solicitud.	b) El sistema actualiza el listado de visitantes.
	2.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y pasa a la acción 2.1 2.4 a) Si los datos introducidos no son los correctos el sistema muestra el mensaje: "Esta palabra no es permitida en este campo" y pasa a la acción 2.1
	3.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y pasa a la acción 3.1 3.4 a) Si los datos introducidos no son los correctos el sistema muestra el mensaje: "Esta palabra no es permitida en este campo" y pasa a la acción 3.1
4.2 a) El trabajador UCI cancela.	
Poscondiciones	Se realiza, cancela o actualiza una solicitud de visita diurna.

Prototipos

Plantilla 1

Plantilla 2

RECIBIDAS

Eilyn Guzmán Rodríguez 2009-06-03 [Solicitud de Visitante Diurna](#)

ENVIADAS

Facultad 1 2009-06-01 10:27:28 [Solicitud de Visitante Diurna](#)

ULTIMAS ENVIADAS

ULTIMAS REVISADAS

B

Plantilla 3

- Solicitud
- Prorrogar
- Notificar
- Registrar
- Historial
- Mis Visitantes

G

EDIT SOLICITUD

Area:

Directivo:

Fecha de Entrada: / /

Motivo:

C

VISITANTES

NombreP	Apellidos	Apellido	CI	
yi	yu	yu	86073003075	editar eliminar

Nombre:

P Apellido:

S Apellido:

Ci:

Organismo:

Plantilla 4

SOLICITUD ENVIADA POR EILYN GUZMÁN RODRÍGUEZ

D

Solicitante:	Eilyn Guzmán Rodríguez
Tipo de Solicitud:	Solicitud de Visitante Diurna
Area:	Facultad 1
Directivo:	Eilyn Guzmán Rodríguez
Estado:	Aprobada
Motivo:	Entrevista de Trabajo
Fecha de Entrada:	2009-06-11
Fecha de Salida:	01-06-09 14:23:48



EILYN/OUZMAN RODRIGUEZ

VISITANTES

Nombre	P Apellido	S Apellido	CI	
HeribertinaRodriguez	Morquilla		80037003045	seleccionar

Caso de Uso.	Gestionar solicitud temporal.
Actor	Trabajador interno.
Resumen.	
El caso de uso comienza cuando el trabajador Interno selecciona alguna de las opciones correspondientes al caso de uso. El usuario podrá realizar una nueva solicitud temporal y a partir del listado de las <i>Solicitudes Enviadas</i> que le será mostrado, podrá modificar o cancelar una solicitud del mismo, pudiendo ver también los detalles de cada solicitud en el listado <i>Últimas Enviadas</i> .	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias.	R3, R3.1,R3.1.1, R3.1.1.1, R3.1.2, R3.1.2.1, R3.3, R3.2, R3.2.1, R3.2.1.1, R3.2.2, R3.2.2.1, R3.3, R3.3.1, R3.3.1.1, R 3.3.2, R3.4, R3.4.1
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. El trabajador interno accede a la página principal.	1.1 El sistema le muestra el listado de las <i>Solicitudes Enviadas</i> que son las que están pendientes de aprobación y la posibilidad de modificar o cancelar cualquiera de estas solicitudes, así como de realizar una nueva solicitud.
1.2. El trabajador interno selecciona una de las	

Página 107

opciones. Si selecciona <i>Solicitud Temporal</i> (A) (ver sección Nueva Solicitud Temporal y Plantilla 1) Si selecciona <i>Modificar</i> (B) (ver sección Modificar Solicitud Temporal y Plantilla 2 y 3) Si selecciona <i>Cancelar</i> (C) (ver sección Cancelar Solicitud Temporal) Si selecciona <i>Detalles</i> (D) (ver sección Detalles solicitud Temporal y plantilla 4).	
Sección Nueva Solicitud Temporal	
	2.1 Se le muestran todos los campos que debe llenar para la solicitud (E) (área, fecha de entrada, fecha de salida, motivos, edificio y cantidad de días) así como la posibilidad de insertar todos los visitantes que desee (nombre del visitante, carnet de identidad, id organismo, sexo y parentesco). Permite insertar visitantes que hayan sido registrados con anterioridad. (F)
2.2. Introduce los datos y envía	2.3 El sistema verifica que estén todos los campos llenos. 2.4 El sistema valida los datos y actualiza.
Sección Modificar Solicitud Temporal	
	3.1 El sistema muestra todos los campos con la información de la solicitud escogida y la posibilidad de modificarlos.(G)
3.2. El trabajador interno modifica los datos que desea y los envía.	3.3 El sistema verifica que estén todos los campos llenos. 3.4 El sistema valida los datos y actualiza.
Sección Cancelar Solicitud Temporal	
	4.1 Muestra el mensaje:"Realmente desea cancelar la solicitud".
4.2. El trabajador interno acepta.	4.3 El sistema actualiza las solicitudes
Sección Detalles Solicitud Temporal	

	5.1 Se le muestra toda la información de la solicitud.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema.
1.2 a) Seleccionar <i>Eliminar</i> si desea eliminar un visitante de la solicitud.	b) El sistema actualiza el listado de visitantes.
	2.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y pasa a la acción 2.4
	2.4 a) El sistema muestra el mensaje: "Esta palabra no es permitida en este campo" y pasa a la acción 2.4
	3.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y pasa a la acción 3.1
	3.4a) Si los datos introducidos no son los correctos el sistema muestra el mensaje: "Esta palabra no es permitida en este campo" y pasa a la acción 3.1
4.2 b) El trabajador interno cancela.	
Poscondiciones	Se realiza, cancela o actualiza una solicitud de visita.

Prototipos

Plantilla 1

The screenshot shows a web application interface for creating a temporary visit request. On the left is a vertical sidebar menu with items: Solicitud, Prorrogar, Notificar, Registrar, Historial, and Mis Visitantes. A callout box 'A' points to the 'Solicitud' item. The main content area is titled 'NUEVA SOLICITUD TEMPORAL' and contains a form with the following fields:

- Area: A dropdown menu with 'Seleccione' as the current selection.
- Directivo: A dropdown menu with 'Rosmel Alvarez Monzon' as the current selection.
- Fecha de Entrada: A date picker with dropdowns for month, day, and year, and a '...' button.
- Fecha de Salida: A date picker with dropdowns for month, day, and year, and a '...' button.
- Motivo: A large empty text input field.
- Apartamento: A text input field.

 Below the form is a section titled 'VISITANTES' with the text 'No existen registros'. To the right of this section are two callout boxes, 'E' and 'F', pointing to the '+ Nuevo', '+ Buscar', and '+ Filtrar' buttons. At the bottom of the interface, there is a footer: '©2009 : Control de Visitas: Todos los derechos reservados'.

Plantilla 2

RECIBIDAS

Eilyn Guzmán Rodríguez 2009-06-03 [Solicitud de Visitante Diurna](#)

Eilyn Guzmán Rodríguez 2009-06-02 [Solicitud Temporal](#)

ENVIADAS

Facultad 1 2009-06-01 10:27:28 [Solicitud de Visitante Diurna](#)

Residencia 2009-06-01 11:11:38 [Solicitud Temporal](#)

B

ULTIMAS ENVIADAS

ULTIMAS REVISADAS

Plantilla 3

EDIT SOLICITUD

Area

Directivo

Fecha de Entrada

Fecha de Salida

Motivo

Apartamento

G

[Eliminar](#)

C

VISITANTES

Nombre	P ApellidoS	Apellido	CI	
Heribertina	guzman	Morquilla	76073003075	editar eliminar

[+](#) Nuevo [+](#) Buscar [+](#) Filtrar

Plantilla 4

SOLICITUD ENVIADA POR EILYN GUZMÁN RODRÍGUEZ	
Solicitante:	Eilyn Guzmán Rodríguez
Tipo de Solicitud:	Solicitud al Hotelito
Area:	Hotelito
Directivo:	Rosmel Alvarez Monzon
Estado:	Aprobada
Motivo:	Proyecto Productivo
Fecha de Entrada:	2009-06-03
Fecha de Salida:	2009-06-19
Fecha de Salida:	01-06-09 13:46:46




VISITANTES				
Nombre	P	ApellidoS	Apellido	CI
HeribertinaRodriguez		Morquilla		65073003075

Caso de Uso.	Gestionar Solicitud al Hotelito.
Actor	Directivo
Resumen	
El caso de uso comienza cuando el Directivo selecciona alguna de las opciones correspondientes al caso de uso. El usuario podrá realizar una nueva solicitud al hotelito y a partir del listado de las <i>Solicitudes Enviadas</i> que le será mostrado, podrá modificar o cancelar una solicitud del mismo, pudiendo ver también los detalles de cada solicitud.	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R4, R4.1,R4.1.1, R4.1.1.1, R4.1.2, R4.1.2.1, R4.3, R4.2, R4.2.1, R4.2.1.1, R4.2.2, R4.2.2.1, R4.3, R4.3.1, R4.3.1.1, R 4.3.2, R4.4, R4.4.1
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. El directivo accede a la página principal del sistema.	1.1 El sistema le muestra el listado de las <i>Solicitudes Enviadas</i> que son las que están pendientes de aprobación y la posibilidad de modificar o cancelar cualquiera de estas solicitudes, así como de realizar una nueva solicitud.

<p>1.2. El Directivo selecciona una de las opciones.</p> <p>Si selecciona <i>Solicitud - Hotelito</i> (A) (ver sección Nueva Solicitud Hotelito y Plantilla 1).</p> <p>Si selecciona <i>Modificar</i> (B) (ver sección Modificar Solicitud Hotelito y Plantilla 2 y 3).</p> <p>Si selecciona <i>Cancelar</i> (C) (ver sección Cancelar Solicitud Hotelito).</p> <p>Si selecciona <i>Detalles</i> (D) (Ver sección <i>Detalles Solicitud Hotelito</i>).</p>	
Sección Nueva Solicitud Hotelito	
	<p>2.1 Se le muestran todos los campos que debe llenar para la solicitud (E), (Fecha Entrada, Fecha Salida, Sexo, Proyecto, Misión o Evento por el que se hospeda) así como la posibilidad de insertar todos los visitantes que desee (Nombre del visitante, Carnet de identidad). Permite insertar visitantes que hayan sido registrados con anterioridad. (F)</p>
<p>2.2. El Directivo ingresa los datos y los envía.</p>	<p>2.3 El sistema verifica que estén todos los campos llenos.</p> <p>2.4 El sistema valida los datos y actualiza.</p>
Sección Modificar Solicitud Hotelito.	
	<p>3.1 El sistema muestra todos los campos con la información de la solicitud escogida y la posibilidad de modificarlos.(G)</p>
<p>3.2. El directivo modifica los datos y los envía.</p>	<p>3.3 El sistema verifica que estén todos los campos llenos.</p> <p>3.4 El sistema valida los datos y actualiza.</p>
Sección Cancelar Solicitud Hotelito	
	<p>4.1 Muestra el mensaje: "Esta seguro que desea cancelar la solicitud"</p>
<p>4.2. El directivo acepta.</p>	<p>4.3 El sistema actualiza las solicitudes.</p>
Sección Detalles Solicitud Hotelito.	
	<p>5.1 El sistema le muestra todos los detalles de la solicitud que seleccionó.</p>

Flujo alternativo de eventos	
Acción del actor	Respuesta del sistema.
1.2 a) Seleccionar <i>Eliminar</i> si desea eliminar un visitante de la solicitud.	b) El sistema actualiza el listado de visitantes.
	2.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." Y pasa a la acción 2.1 2.4 a) Si los datos introducidos no son los correctos el sistema muestra el mensaje: "Esta palabra no es permitida en este campo" y pasa a la acción 2.1
	3.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." Y pasa a la acción 3.1 3.4 a) Si los datos introducidos no son los correctos el sistema muestra el mensaje: "Esta palabra no es permitida en este campo" y pasa a la acción 3.1
4.2 a) El trabajador UCI cancela.	
Poscondiciones	Se realiza, cancela o actualiza una solicitud de visita al hotelito.

Prototipos

Plantilla 1

Plantilla 2

RECIBIDAS

Eilyn Guzmán Rodríguez 2009-06-03 [Solicitud de Visitante Diurna](#)

Eilyn Guzmán Rodríguez 2009-06-02 [Solicitud Temporal](#)

ENVIADAS

Facultad 1 2009-06-01 10:27:28 [Solicitud de Visitante Diurna](#)

Residencia 2009-06-01 11:11:38 [Solicitud Temporal](#)

Hotelito 2009-06-01 11:30:46 [Solicitud al Hotelito](#)

B

ULTIMAS ENVIADAS

ULTIMAS REVISADAS

Plantilla 3

EDIT SOLICITUD

G

Area

Directivo

Fecha de Entrada

Fecha de Salida

Proyecto o Misión

C

[Eliminar](#)

VISITANTES

Nombre	P ApellidoS	Apellido	CI	
HeribertinaRodriguez	Morquilla		65073003075	editar eliminar

[+](#) Nuevo [+](#) Buscar [+](#) Filtrar

Caso de Uso.

Aprobar/denegar solicitudes.

Actor	Directivo.
Resumen	
El caso de uso comienza cuando el directivo accede a su página principal y selecciona en el listado de <i>Solicitudes Recibidas</i> la solicitud que desea revisar. El sistema muestra los detalles de cada solicitud, así como también le muestra los botones Aceptar y Denegar (en este caso debe exponer los motivos por los cuáles deniega la solicitud). El caso de uso concluye cuando el Directivo culmina una operación.	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R6, R6.1, R6.1.1, R6.1.2, R6.1.3, R6.2, R6.2.1
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1 El Directivo accede a su página principal.	1.1 El sistema muestra un listado de las solicitudes recibidas, que están pendientes de aprobación. (A)
1.2 El Directivo selecciona la solicitud que desea aprobar.	1.3 El sistema muestra los detalles de la solicitud y los botones "Aprobar" y "Denegar". (B)
1.4 El Directivo presiona el botón Aprobar.	1.3 El sistema muestra el mensaje "Solicitud aprobada con éxito" y actualiza.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema
1.4 a) El Directivo sitúa el cursor encima del botón denegar y el sistema desplegará el campo "Motivos" que el Directivo debe llenar.	b) El sistema verifica que este campo no se encuentre vacío. c) El sistema verifica que los datos sean válidos. d) El sistema muestra el mensaje "Solicitud denegada con éxito" y actualiza.
Poscondiciones	
Prototipo	
Plantilla 1	

RECIBIDAS

Eilyn Guzmán Rodríguez	2009-06-03	Solicitud de Visitante Diurna
Eilyn Guzmán Rodríguez	2009-06-02	Solicitud Temporal

A

ENVIADAS

Facultad 1	2009-06-01 10:27:28	Solicitud de Visitante Diurna
Residencia	2009-06-01 11:11:38	Solicitud Temporal
Hotelito	2009-06-01 11:30:46	Solicitud al Hotelito

ULTIMAS ENVIADAS

ULTIMAS REVISADAS

Plantilla 2

SOLICITUD ENVIADA POR EILYN GUZMÁN RODRÍGUEZ

Solicitante:	Eilyn Guzmán Rodríguez
Tipo de Solicitud:	Solicitud de Visitante Diurna
Area:	Facultad 1
Directivo:	Eilyn Guzmán Rodríguez
Estado:	Pendiente
Motivo:	Entrevista de trabajo
Fecha de Entrada:	2009-06-03



VISITANTES

Nombre	P	ApellidoS	Apellido	CI	
HeribertinaRodriguez		Morquilla		80073003075	seleccionar

Aprobar

Denegar

B

Caso de Uso.

Registrar entrada/salida de visitantes.

Actor	Técnico SP.
Resumen	
El caso de uso se inicia cuando el visitante accede a la garita y solicita entrar a la universidad. Se mostrará la opción de <i>Registrar Nueva Visita</i> en caso de que la visita no esté registrada y sea gestionada en el momento. Permite también <i>Registra Entrada/ Salida</i> de Visitas programadas.	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R7, R7.1, R7.1.1, R7.1.2, R7.1.3, R7.1.4, R7.1.5
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. El Técnico SP Selecciona la opción Registrar Entrada/Salida de Visita. Si selecciona <i>Registrar Imprevista (A)</i> (Ver Sección Registrar Nueva Visita y Plantilla 1). Si selecciona <i>Registrar Entrada/Salida de Vista</i> (Ver Sección Registrar Entrada/Salida de Vista y Plantilla 2 y 3).	
Sección Registrar Nueva Visita	
	2.1 El sistema muestra una interfaz con los campos requeridos en dicha operación. 1) Nombre del visitante. 2) No carnet de Identidad. 3) Fecha de entrada. 4) Organismo. 5) Motivos de la solicitud. 6) Estado de la visita.
2.2 El Técnico SP introduce los datos del visitante 2.3 El Técnico SP acepta.	2.4 El sistema verifica que estén todos los campos llenos. 2.5 El sistema verifica validez de los datos.
Sección Registrar Entrada/Salida de vista	
3.1 El usuario introduce el número de carnet de identidad del visitante.	3.2 El sistema muestra los datos del visitante encontrado con la posibilidad de registrar la entrada.

3.3 Registra la entrada introduciendo los datos necesarios.	3.4 El sistema guarda la hora en que se realizó la visita y el estado del visitante pasa a ser "En curso" (B)
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema
	2.5 a) El sistema muestra el mensaje: "Esta palabra no es permitida en este campo". b) El sistema pasa a la acción 2.1
	3.4 a) En caso de que el visitante tenga como estado "En curso" se mostrarán los campos para registrar la salida con la opción de introducir las observaciones de la visita y el botón <i>Registrar la salida</i> (C) y plantilla 4. El sistema guarda la hora y el estado de la misma pasa a "Terminada".
Poscondiciones	La visita queda registrada en la base de datos con el estado "en curso" si entró o "terminada" si salió.

Prototipo

Plantilla 1

The screenshot shows a web application interface. On the left is a vertical sidebar menu with the following items: Solicitud, Prorrogar, Notificar, Registrar, Historial, and Mis Visitantes. The 'Mis Visitantes' item is highlighted, and a sub-menu is visible with the options: Entrada, Salida, and Imprevista. A callout bubble with the letter 'A' points to the 'Imprevista' option. The main content area is titled 'NUEVA SOLICITUD DIURNA' and contains a form with the following fields: 'Area' (a dropdown menu with 'Seleccione' as the current selection), 'Nombre' (a dropdown menu with 'Rosmel Alvarez Monzon' as the current selection), 'Fecha de Entrada' (a date selection widget with three dropdowns and a calendar icon), and 'Motivo' (a large text input area). Below the form is an 'Aceptar' button. At the bottom of the main area, there is a section titled 'VISITANTES' with the text 'No existen registros' and three buttons: '+ Nuevo', '+ Buscar', and '+ Filtrar'.

Plantilla 2

Solicitud	
Prorrogar	
Notificar	
Registrar	Entrada
Historial	Salida
Mis Visitantes	Imprevista

REGISTRAR ENTRADA

Ask for a job

B

Plantilla 3

REGISTRAR ENTRADA

800

1

Heribertina 2009-06-03 80073003075

Observaciones

B

Plantilla 4

REGISTRAR SALIDA

800

1

Heribertina 2009-06-03 80073003075

Observaciones

C

Caso de Uso.	Autenticar Usuario
Actor	Usuario.

Resumen.	
El caso de uso comienza cuando el usuario abre el navegador para entrar al sistema, y este pide que se autentique con usuario y contraseña del dominio UCI.	
Precondiciones.	Que el sistema esté disponible.
Referencias.	R1, R1.1, R1.1.1, R1.2, R1.2.1, R1.3
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. El usuario accede al sistema.	1.2. El sistema muestra una interfaz de inicio de sección.
2. Introduce usuario y clave del dominio.	2.1. El sistema verifica que estén los campos llenos. 2.2. El sistema verifica que sea usuario del dominio UCI. 2.3. Se configura el sistema con los privilegios correspondientes al usuario autenticado.
Flujos alternos de trabajo	
Acción del actor	Respuesta del negocio
	2.1 Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y el sistema pasa a la acción 1.2.
	2.2 El sistema muestra el mensaje: "Usuario o contraseña son inválidos." y el sistema pasa a la acción 1.2.
Poscondiciones.	El usuario accede al sistema según sus privilegios o le es denegado el acceso.
Prototipo	

Username

Password

Remember

[Forgot your password?](#)

©2009 : Control de Visitas: Todos los derechos reservados

Caso de Uso.	Notificar visitas
Actor	Directivo
Resumen.	
El caso de uso inicia cuando el directivo selecciona la opción <i>Notificar visita</i> . El sistema le mostrará el listado de los tipos de visita que puede notificar. Al seleccionar un tipo de visita el sistema le mostrará los campos requeridos. Estas visitas no requieren de más autorizaciones.	
Precondiciones.	Que el usuario este autenticado en el sistema con el rol y los permisos requeridos para esta funcionalidad.
Referencias.	R5, R5.1, R5.2, R5.2.1, R5.3, R.5.3.1, R5.4
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. El Directivo selecciona la opción <i>Notificar visita</i> . Ver plantilla 1.	1.1. El sistema muestra una interfaz con la posibilidad de seleccionar el tipo de visita que desea notificar.
2. El Directivo selecciona el tipo de visita que desea notificar.	2.1 El sistema le muestra los campos correspondientes a la solicitud seleccionada.
3. El Directivo introduce los datos requeridos por el sistema.	3.1 El sistema verifica que estén todos los campos llenos. 3.2 El sistema verifica validez de los datos y actualiza.

Flujos alternos de trabajo											
Acción del actor	Respuesta del negocio										
	<p>3.1 Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y el sistema pasa a la acción 2.1.</p> <p>3.2 Si los datos no son válidos el sistema muestra el mensaje "Esta palabra no es permitida en este campo" y el sistema pasa a la acción 2.1.</p>										
Poscondiciones.	El usuario notifica la visita deseada sin la necesidad de más autorizaciones. Estado de la visita "Pendiente".										
Prototipo											
Plantilla 1											
<p>The screenshot displays a web application interface. On the left is a vertical sidebar menu with the following items: 'Solicitud', 'Prorrogar', 'Notificar', 'Registrar', 'Historial', and 'Mis Visitantes'. The main content area is divided into three sections:</p> <ul style="list-style-type: none"> RECIBIDAS: A blue header bar above a table with one row: <table border="1"> <tr> <td>Diurna</td> <td>guez</td> <td>2009-06-02</td> <td>Solicitud Temporal</td> </tr> </table> ULTIMAS ENVIADAS: A blue header bar above a table with one row: <table border="1"> <tr> <td>Eilyn Guzmán Rodríguez</td> <td>01-06-09 11:38:43</td> <td>Aprobada</td> </tr> </table> ULTIMAS REVISADAS: A blue header bar above a table with one row: <table border="1"> <tr> <td>Eilyn Guzmán Rodríguez</td> <td>01-06-09 11:38:43</td> <td>Aprobada</td> </tr> </table> 		Diurna	guez	2009-06-02	Solicitud Temporal	Eilyn Guzmán Rodríguez	01-06-09 11:38:43	Aprobada	Eilyn Guzmán Rodríguez	01-06-09 11:38:43	Aprobada
Diurna	guez	2009-06-02	Solicitud Temporal								
Eilyn Guzmán Rodríguez	01-06-09 11:38:43	Aprobada									
Eilyn Guzmán Rodríguez	01-06-09 11:38:43	Aprobada									

Caso de Uso	Mostrar Historial
Actor	Trabajador UCI
Resumen	
El caso de uso se inicia cuando el actor entra al sistema. El sistema le mostrará un listado (<i>Ultimas Enviadas</i>) con todas las solicitudes que ha realizado a las diferentes áreas así como el estado correspondiente de cada	

una. También brindará la posibilidad de ver los <i>Detalles</i> (B) de cada una.																			
Precondiciones	El actor debe autenticarse para comprobar los permisos y acceder al servicio.																		
Referencias	R8, R8.1, R8.1.1, R8.2, R8.2.1																		
Flujo normal de eventos																			
Acción del actor	Respuesta del sistema																		
1. El Trabajador UCI introduce el número de solicitudes que desea ver.	1.1 El sistema muestra el listado de la cantidad de solicitudes realizadas con anterioridad.																		
Sección Detalles																			
	2.1 Muestra los detalles de la solicitud seleccionada.																		
Poscondiciones	El usuario obtiene la información solicitada.																		
Prototipo																			
<table border="1"> <tr><td>Solicitud</td></tr> <tr><td>Prorrogar</td></tr> <tr><td>Notificar</td></tr> <tr><td>Registrar</td></tr> <tr><td>Historial</td></tr> <tr><td>Mis Visitantes</td></tr> </table>	Solicitud	Prorrogar	Notificar	Registrar	Historial	Mis Visitantes	<div style="text-align: center; background-color: #0070C0; color: white; padding: 5px; border-radius: 10px; margin-bottom: 10px;">TODAS LAS SOLICITUDES HISTORIAL</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Eilyn Guzmán Rodríguez</td> <td style="padding: 5px;">2009-06-01 11:11:38</td> <td style="padding: 5px;">Solicitud Temporal</td> <td style="padding: 5px;">comun</td> </tr> <tr> <td style="padding: 5px;">Rosmel Alvarez Monzon</td> <td style="padding: 5px;">2009-06-01 11:30:46</td> <td style="padding: 5px;">Solicitud al Hotelito</td> <td style="padding: 5px;">comun</td> </tr> <tr> <td style="padding: 5px;">Eilyn Guzmán Rodríguez</td> <td style="padding: 5px;">2009-06-01 10:27:28</td> <td style="padding: 5px;">Solicitud de Visitante Diurna</td> <td style="padding: 5px;">comun</td> </tr> </table> <p style="margin-top: 10px;"><i>Historial (3)</i></p>	Eilyn Guzmán Rodríguez	2009-06-01 11:11:38	Solicitud Temporal	comun	Rosmel Alvarez Monzon	2009-06-01 11:30:46	Solicitud al Hotelito	comun	Eilyn Guzmán Rodríguez	2009-06-01 10:27:28	Solicitud de Visitante Diurna	comun
Solicitud																			
Prorrogar																			
Notificar																			
Registrar																			
Historial																			
Mis Visitantes																			
Eilyn Guzmán Rodríguez	2009-06-01 11:11:38	Solicitud Temporal	comun																
Rosmel Alvarez Monzon	2009-06-01 11:30:46	Solicitud al Hotelito	comun																
Eilyn Guzmán Rodríguez	2009-06-01 10:27:28	Solicitud de Visitante Diurna	comun																

Caso de Uso.	Solicitar prórroga de hospedaje en el hotelito
Actor	Directivo
Resumen.	
El caso de uso comienza cuando el actor selecciona la opción <i>Solicitud de Prórroga</i> (A). El sistema le muestra el listado del Proyecto, Misión o Evento por el que se haya hospedado la visita. Al seleccionar uno de estos eventos el sistema le muestra el listado de visitas que fueron realizadas para el mismo y la posibilidad de solicitar una nueva fecha de salida para todos o solo para algunos .El sistema brindará la posibilidad de revisar los <i>Detalles</i> (B) de cualquiera de las visitas.	
Precondiciones	El actor debe autenticarse para comprobar los permisos y acceder al servicio y debe poseer al menos un visitante en al menos un evento.

Referencias	R14, R14.1, R14.2, R14.3,R14.4, R14.4.1,R14.5
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción <i>Solicitud de Prórroga</i> .	1.1 El sistema muestra el listado de los eventos para los cuales posee visitas en la universidad.
2. Selecciona el evento que posee al visitante para el cual solicita la prórroga.	2.1 Muestra el listado de los visitantes que posee el Directivo para ese evento.
3. Selecciona el o los visitante para los que solicita prórroga e introduce los datos requeridos por el sistema y acepta.	3.1 El sistema verifica que no existan campos vacios. 3.2 El sistema verifica validez de los datos y actualiza.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema
3 El Directivo no selecciona ningún visitante y acepta.	3 a) El sistema muestra el mensaje “Debe seleccionar al menos un visitante” y el sistema pasa a la acción 2.1.
	3.1 Si los campos no están llenos el sistema muestra el mensaje: “Campos incompletos.” y el sistema pasa a la acción 2.1. 3.2 Si los datos no son válidos el sistema muestra el mensaje “Esta palabra no es permitida en este campo” y el sistema pasa a la acción 2.1
Poscondiciones	Queda solicitada la prórroga de hospedaje en el hotelito de la universidad.
Prototipo	

Caso de Uso.	Solicitar prórroga para visitas temporales.
Actor	Trabajador interno.
Resumen.	El caso de uso comienza cuando el actor selecciona la opción <i>Solicitud de Prórroga</i> (A). El sistema le muestra el listado de las visitas temporales que se encuentran en la universidad (B) y que han sido solicitadas por él. Al seleccionar uno de estos visitantes o varios visitantes, el sistema le brinda la posibilidad de solicitar una nueva

	fecha de salida para los seleccionados y los motivos por los cuales solicita la prórroga. (C)
Precondiciones	El actor debe autenticarse para comprobar los permisos y acceder al servicio y debe poseer al menos una visita temporal en la universidad.
Referencias	R13.1 ,R13.2, R13.3, R13.3.1, R13.4
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Selecciona la opción <i>Solicitud de Prórroga</i> .	1.1 El sistema muestra el listado de los visitantes que se encuentran en la universidad, que han sido solicitados por el actor.
2. Selecciona el o los visitante para los que solicita prórroga e introduce los datos requeridos por el sistema y acepta.	2.1 El sistema verifica que no existan campos vacíos. 2.2 El sistema verifica validez de los datos y actualiza.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema
2 El Directivo no selecciona ningún visitante y acepta.	2 a) El sistema muestra el mensaje “Debe seleccionar al menos un visitante” y el sistema pasa a la acción 1.1.
	2.1 Si los campos no están llenos el sistema muestra el mensaje: “Campos incompletos.” y el sistema pasa a la acción 1.1. 2.2 Si los datos no son válidos el sistema muestra el mensaje “Esta palabra no es permitida en este campo” y el sistema pasa a la acción 1.1
Poscondiciones	Queda realizada la solicitud de prórroga de la, o las visitas temporales.
Prototipo	

Sistema de Reservación de Visitas

[Solicitud Temporal](#)

[Prórroga](#)

[Solicitud Diurna](#)

[Historial](#)

[Cerrar Sección](#)

	Nombre del visitante	Fecha de Entrada	Fecha de Salida	Detalles
<input type="checkbox"/>	Irina Fuentes Viñas	15/5/2009	18/5/2009	Detalles
<input type="checkbox"/>	Eilyn Guzman Rodriguez	15/5/2009	18/5/2009	Detalles
<input type="checkbox"/>	Heribertina Rodriguez	15/5/2009	18/5/2009	Detalles

Nueva Fecha de Salida:

Motivos de solicitud:

Caso de Uso.	Reporte de SP.
Actor	Especialista
Resumen	
<p>El caso de uso se inicia cuando el actor accede al sistema y al autenticarse selecciona la opción <i>Realizar Búsqueda</i>. El sistema le brindará la posibilidad de realizar la búsqueda de cualquier solicitud, solicitante, así como visitante que haya sido registrado con anterioridad en el sistema. Esta búsqueda se realiza bajo los criterios de búsqueda: solicitante, solicitante, fecha de entrada o de salida, organismo al que pertenece o número de carnet de identidad. Una vez obtenido el resultado, podrá ver los detalles completos del mismo.</p>	
Poscondiciones	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R9,R9.1, R9.2
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El Especialista selecciona la opción <i>Realizar Búsqueda</i> .	1.1 El sistema muestra los posibles criterios de búsqueda.
2. El Especialista selecciona el criterio por el que desea buscar. Introduce el dato y acepta.	2.1 El sistema muestra los campos de datos correspondientes a la selección de criterio.
3. El Especialista introduce los datos.	3.1 El sistema verifica que no existan campos vacíos.
	3.2 El sistema verifica que los datos sean válidos.
	3.3 El sistema realiza la búsqueda y muestra los

	datos solicitados.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema
	<p>3.1 Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." y el sistema pasa a la acción 1.1.</p> <p>3.2 Si los datos no son válidos el sistema muestra el mensaje "Esta palabra no es permitida en este campo" y el sistema pasa a la acción 1.1</p> <p>3.3 En caso de no encontrar los datos solicitados muestra un mensaje de notificación.</p>
Poscondiciones	Se obtienen los datos buscados.

Caso de Uso.	Gestionar usuarios.
Actor	Administrador.
Resumen	
El caso de uso comienza cuando el actor luego de autenticarse selecciona la opción <i>Gestionar usuario</i> permitiéndosele adicionar, modificar y eliminar cada usuario del sistema.	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias.	R10, R10.1, R10.1.1, R10.1.2, R10.1.10, R10.1.10.1, R10.1.4, R10.1.4.1, R10.1.5, R10.2, R10.2.1, R10.2.1.1, R10.2.2, R10.10, R10.10.1, R10.10.1.1, R10.10.10.1.2, R10.10.2
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. Selecciona la opción <i>Gestionar usuarios</i>	1.1 El sistema muestra las opciones Modificar, Eliminar y Adicionar un usuario.
<p>1.2 El Administrador selecciona una de las opciones.</p> <p>Si selecciona <i>Modificar Usuario</i> (A) (ver sección Modificar Usuario y Plantilla 1)</p> <p>Si selecciona <i>Eliminar Usuario</i> (B) (ver sección Eliminar</p>	

Usuario y Plantilla 2) Si selecciona <i>Adicionar Usuario</i> (C) (ver sección Adicionar Usuario y Plantilla 3)	
Sección Modificar Usuario	
	2.1 Muestra el listado de los usuarios
2.2 Selecciona el que desea modificar.	2.3 Muestra los campos con la información anterior dando la posibilidad de modificar la misma seleccionando nuevo rol y nuevos permisos.
2.4 Modifica la información y la envía.	2.5 El sistema verifica que estén todos los campos llenos. 2.6 El sistema verifica validez de los datos.
Sección Eliminar Usuario	
	4.1 Muestra el listado de los usuarios registrados en el sistema.
4.2 Selecciona el(los) que desea eliminar y acepta	4.3 Elimina el(los) usuario(s).
Sección Adicionar Usuario	
	3.1 Muestra los campos que deben ser llenados.
3.2 Llena los campos y envía la información.	3.3 Verifica que no existan campos vacíos y adiciona.
Poscondiciones	Quedan actualizados los usuarios seleccionados con sus nuevos roles y permisos, así como los eliminados.
Prototipo	
Plantilla 1	

EDITAR USUARIO "IFUENTES"

Usuario uci

Usuario

Solapin



Permisos y roles

Is active

Is super admin

Groups

Permissions

 Delete  Cancel

Plantilla 2

USUARIOS (NO EDITAR DIRECTIVOS AQUI)

<input type="checkbox"/>	Usuario	Created at	Updated at	Last login	Actions
<input type="checkbox"/>	yagodinez	1 June 2009 12:33			 Edit  Delete
<input type="checkbox"/>	ltravieso	1 June 2009 12:41	1 June 2009 12:41		 Edit  Delete
<input type="checkbox"/>	ifuentes	1 June 2009 3:15	1 June 2009 13:31	1 June 2009 1:31	 Edit  Delete
<input type="checkbox"/>	rosmel	1 June 2009 3:14	1 June 2009 13:36	1 June 2009 1:36	 Edit  Delete
<input type="checkbox"/>	eguzman	1 June 2009 10:13	1 June 2009 14:22	1 June 2009 2:22	 Edit  Delete
<input type="checkbox"/>	admin	1 June 2009 3:00	1 June 2009 14:34	1 June 2009 2:34	 Edit  Delete
<input type="checkbox"/>	dceperor	1 June 2009 10:19	1 June 2009 10:19		 Edit  Delete
7 results					

Plantilla 3

NUEVO USUARIO

Usuario uci

Usuario	<input type="text"/>
Solapin	<input type="text"/>

Permisos y roles

Is active	<input checked="" type="checkbox"/>
Is super admin	<input checked="" type="checkbox"/>
Groups	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> potestados directivos tecnico_sp </div>
Permissions	<div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> solicitudes_directivos crear_solicitud eliminar_solicitud registrar </div>

Caso de Uso.	Gestionar permisos
Actor	Administrador
Resumen	
El caso de uso comienza cuando el Administrador selecciona la opción <i>Gestionar permisos</i> y se le brinda la posibilidad de <i>Adicionar Permisos</i> y <i>Eliminar Permisos</i>	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R12, R12.1, R12.1.1, R12.1.2, R12.1.3, R12.1.3.1, R12.1.12, R12.2, R12.2.1, R12.2.1.1, R12.2.2
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. Selecciona la opción <i>Gestionar Permisos</i>	1.1. Muestra las opciones: Adicionar permisos y Eliminar permisos.
1.2. Selecciona una de las opciones. Si selecciona <i>Adicionar Permiso</i> (ver sección Adicionar permisos y Plantilla 1). Si selecciona <i>Eliminar Permiso</i> (ver sección Eliminar Permisos y Plantilla 2).	
Sección Adicionar Permiso	
	2.1 Muestra los campos que deben ser llenados.

2.2 Llena los campos y envía la información.	2.3 Verifica que no existan campos vacíos y adiciona.
Sección Eliminar Permiso.	
	3.1 Muestra el listado de los permisos.
3.2 Selecciona los que desea eliminar y acepta	3.3 Elimina el permiso.
Flujos alternos de trabajo	
Acción del actor	Respuesta del sistema
	2.3 a) Si los campos no están llenos el sistema muestra el mensaje: "Campos incompletos." b) El sistema muestra nuevamente los datos introducidos esperando porque el usuario llene los restantes.
Poscondiciones	Quedan actualizados los permisos.

Prototipos









Plantilla 1

NEW PERMISSION

Name	<input style="width: 90%;" type="text"/>
Description	<input style="width: 90%; height: 40px;" type="text"/>
Groups list	<div style="border: 1px solid #ccc; padding: 2px;"> <p>potestados ▲</p> <p>directivos ▼</p> <p>tecnico_sp ▼</p> </div>
Users list	<div style="border: 1px solid #ccc; padding: 2px;"> <p>yagodinez ▲</p> <p>ltravieso ▼</p> <p>ifuentes ▼</p> <p>rosmel ▼</p> <p>eguzman ▼</p> <p>admin ▼</p> <p>dceperor ▼</p> </div>

Cancel

Plantilla 2

PERMISSION LIST					
<input type="checkbox"/>	Name	Description	Created at	Updated at	Actions
<input type="checkbox"/>	solicitudes_directivos		1 June 2009 3:13	1 June 2009 3:13	 Edit  Delete
<input type="checkbox"/>	crear_solicitud				 Edit  Delete
<input type="checkbox"/>	eliminar_solicitud	elimina	1 June 2009 12:38	1 June 2009 12:38	 Edit  Delete
<input type="checkbox"/>	registrar		1 June 2009 13:10	1 June 2009 13:10	 Edit  Delete
4 results					

Caso de Uso.	Gestionar roles
Actor	Administrador
Resumen	
El caso de uso comienza cuando el actor selecciona la opción <i>Gestionar Roles</i> y se le brindan la posibilidad de adicionar roles y eliminar roles.	
Precondiciones.	El actor debe autenticarse para comprobar los permisos y acceder al servicio.
Referencias	R11, R11.1, R11.1.1, R11.1.2, R11.1.3, R11.1.3.1, R11.1.4, R11.1.5, R11.1.6, R11.2, R11.2.1, R11.2.2
Flujo normal de eventos.	
Acción del actor.	Respuesta del sistema.
1. Selecciona la opción <i>Gestionar Roles</i>	1.1 Muestra las opciones: <i>Adicionar Rol</i> y <i>Eliminar Rol</i> .
2. Selecciona una de las opciones. Si selecciona <i>Adicionar Rol</i> (Ver Sección Adicionar Roles y Plantilla 1) Si selecciona <i>Eliminar Rol</i> (Ver Sección Eliminar Roles y Plantilla 2)	
Sección Adicionar Rol	
	2.1 Muestra los campos que deben ser llenados y la lista de permisos.
2.2 Llena los campos y selecciona los permisos. Acepta.	2.3 El sistema verifica que estén todos los campos llenos.

	2.4 El sistema verifica validez de los datos.
Sección Eliminar Rol	
	3.1 Muestra el listado de los roles
3.2 Selecciona el que desea eliminar y acepta	3.3 Elimina rol.
Flujo alternativo de trabajo	
Acción del actor	Respuesta del sistema.
	2.3 a) Si los campos no están llenos el sistema muestra un mensaje de error: "Campos incompletos." b) El sistema pasa a la acción 2.1.3
	2.4 a) Si los campos no están llenos el sistema muestra un mensaje de error: "Campos incompletos." b) El sistema pasa a la acción 2.2.1
Poscondiciones:	Quedan definidos los roles.

Prototipo

Plantilla 1

NEW GROUP

Name

Description

Users list

- itravieso
- ifuentes
- rosmel
- eguzman
- admin
- dceperor

Permissions list

- crear_solicitud
- eliminar_solicitud
- registrar

Plantilla 2

GROUP LIST						
<input type="checkbox"/>	Name	Description	Created at	Updated at	Actions	
<input type="checkbox"/>	potestados	fghfghfghg	1 June 2009 12:39	1 June 2009 12:39	Edit	Delete
<input type="checkbox"/>	directivos				Edit	Delete
<input type="checkbox"/>	tecnico_sp	ghjggjghj			Edit	Delete
3 results						

New

Anexo #4 Diagramas de clases del análisis.

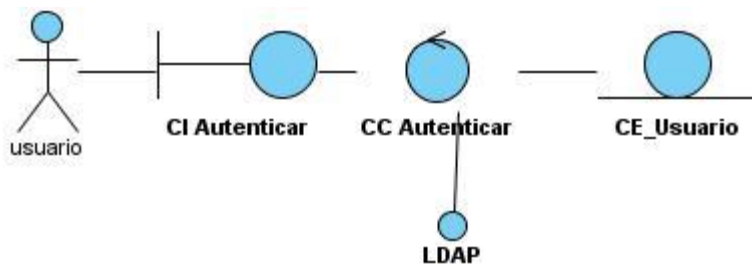


Diagrama de clases del análisis CU Autenticar usuario.

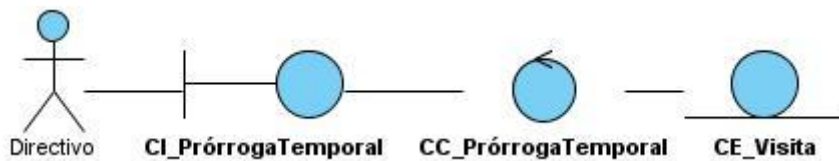


Diagrama de clases del análisis CU Solicitar prórroga para visitas temporales.

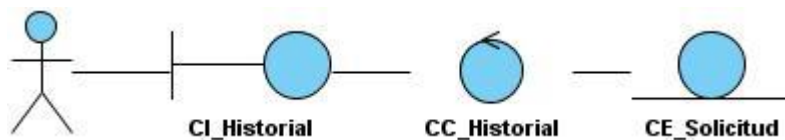


Diagrama de clases del análisis CU Mostrar historial

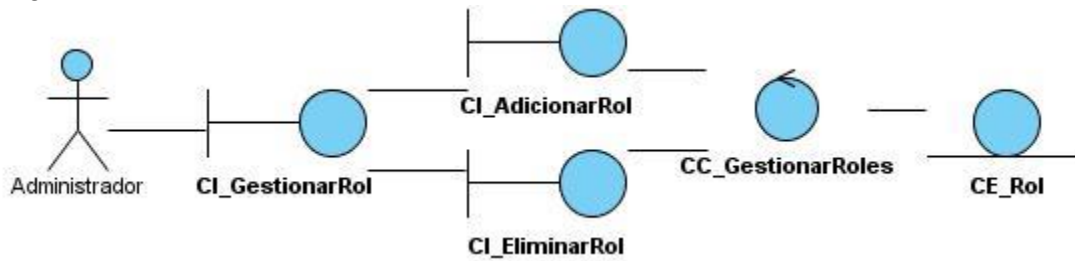


Diagrama de clases del análisis CU Gestionar Rol

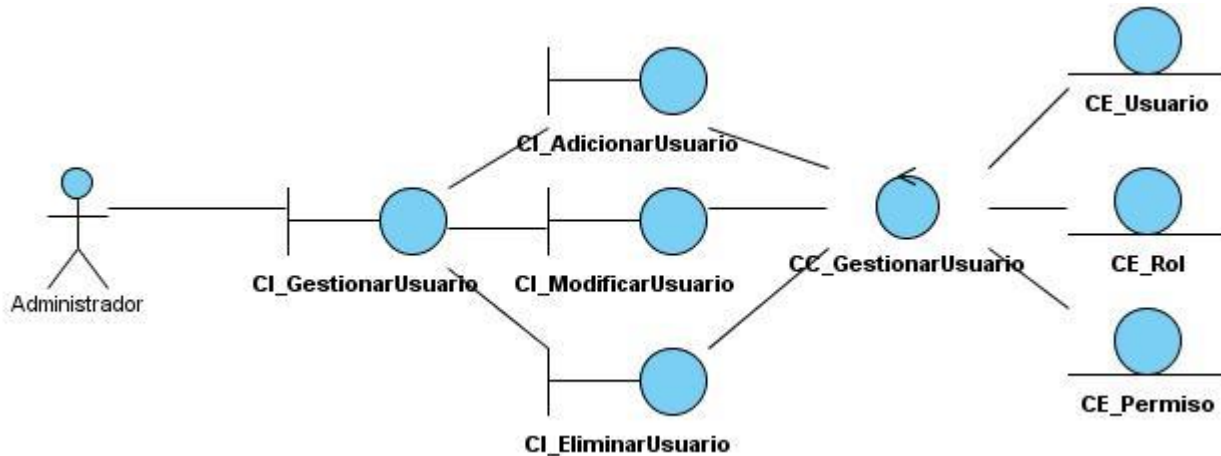


Diagrama de clases del análisis CU Gestionar usuario

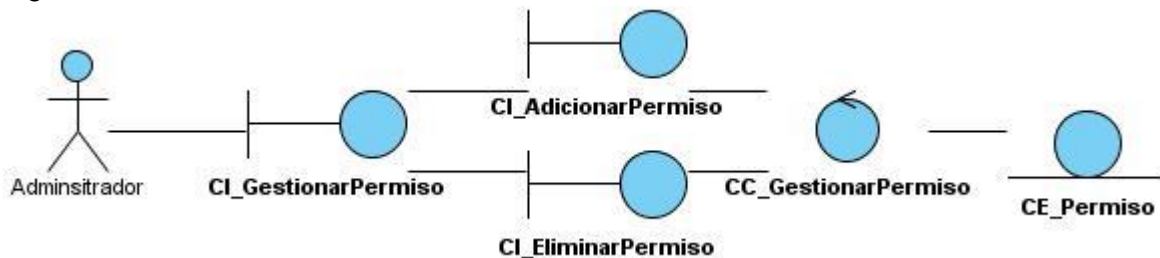


Diagrama de clases del análisis CU Gestionar Permiso

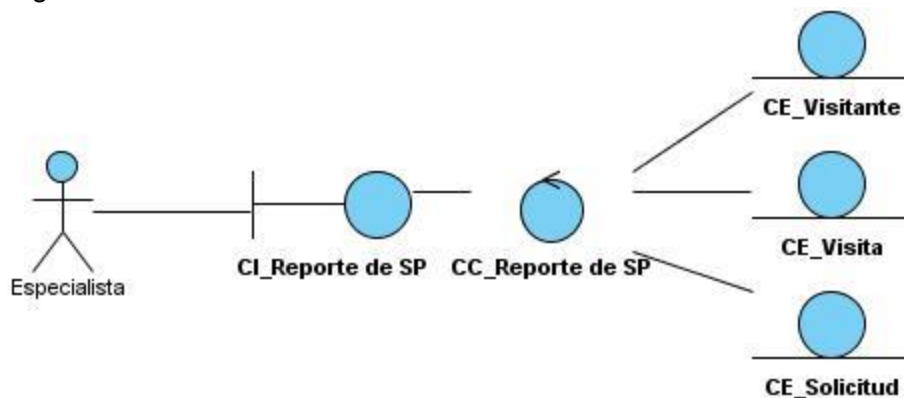


Diagrama de clases del análisis CU Reporte de SP

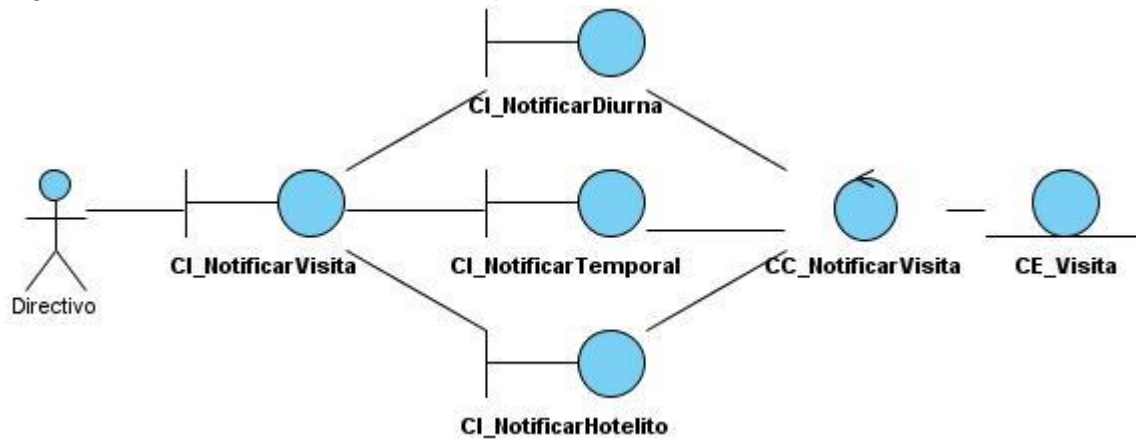


Diagrama de clases del análisis CU Notificar visita

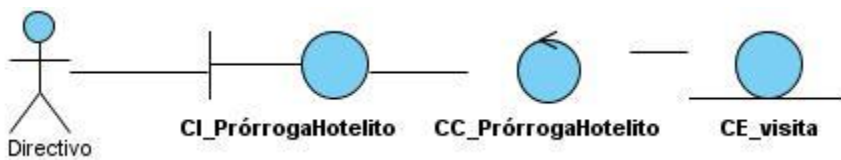


Diagrama de clases del análisis CU Solicitar prórroga para visita del hotelito.

Anexo #5 Diagrama de clases del diseño.

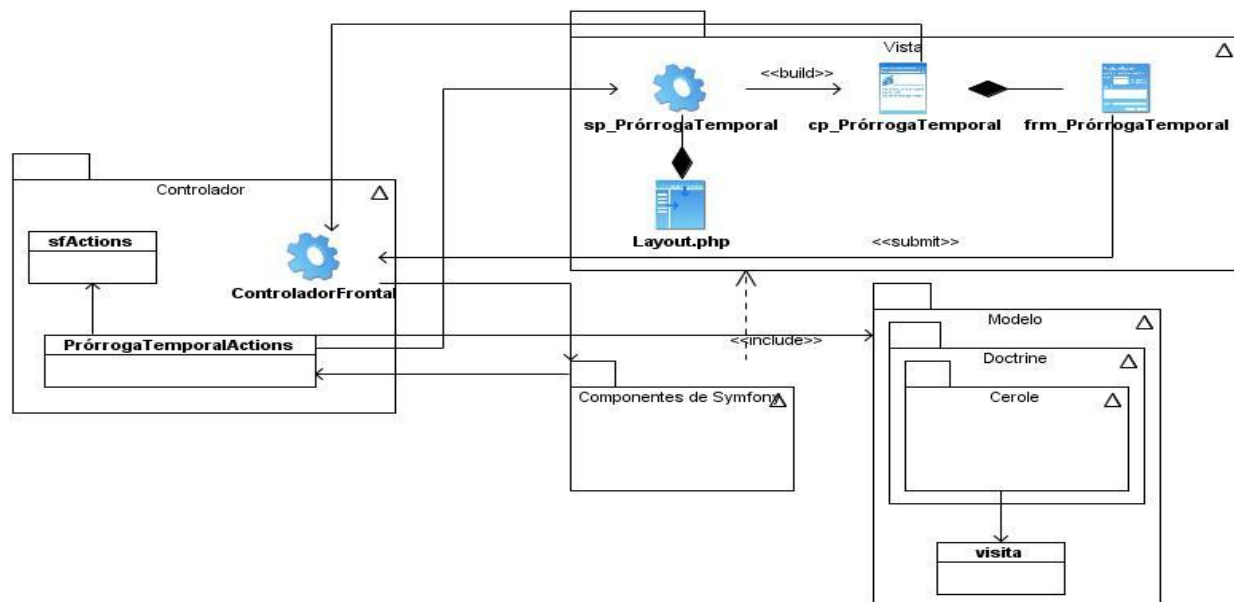


Diagrama de clases del diseño CU Solicitar prórroga para visitas temporales.

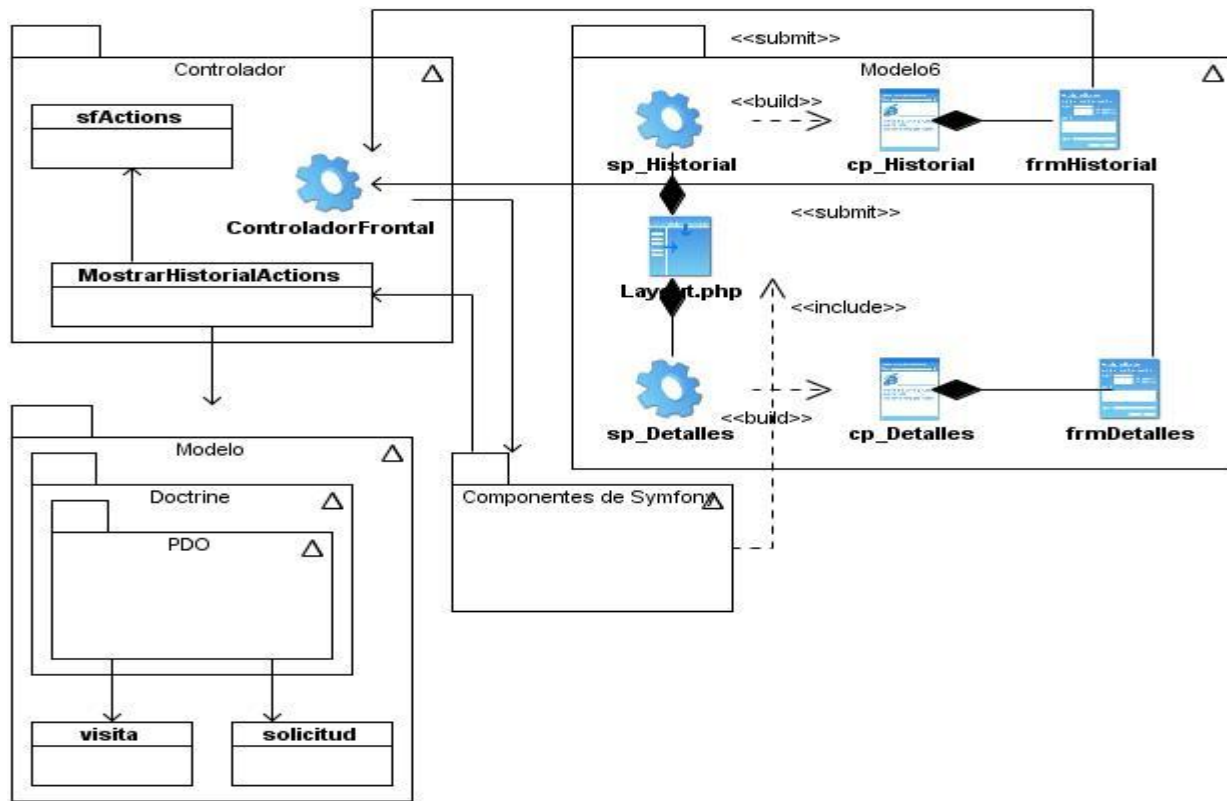


Diagrama de clases del diseño CU Mostrar historial.

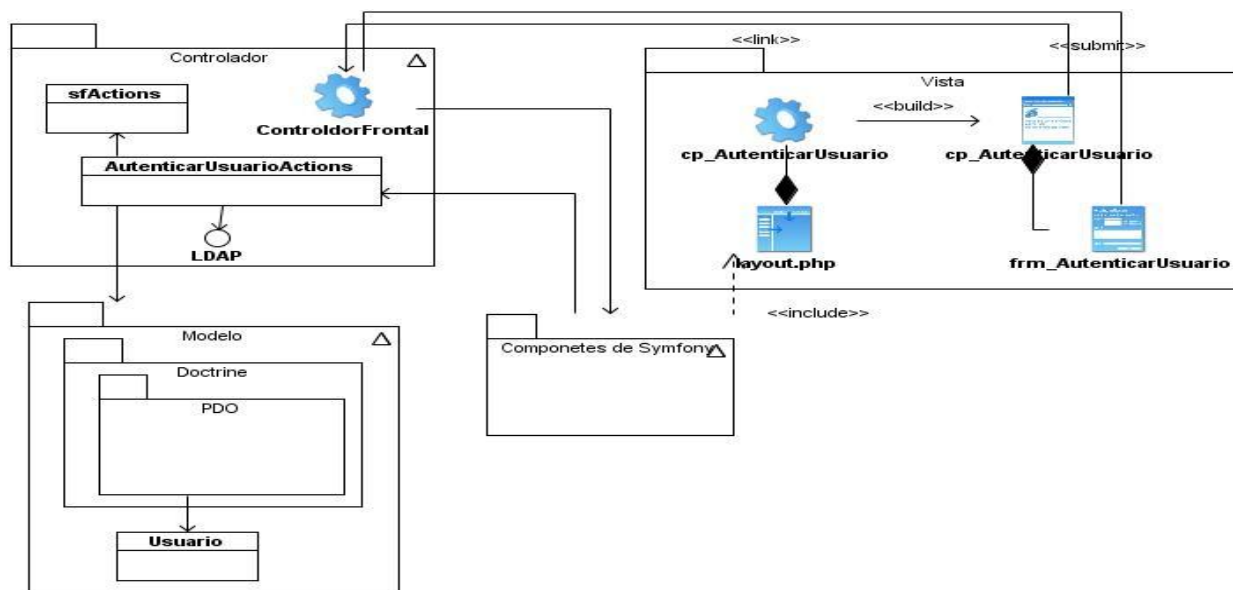


Diagrama de clases del diseño CU Autenticar usuario.

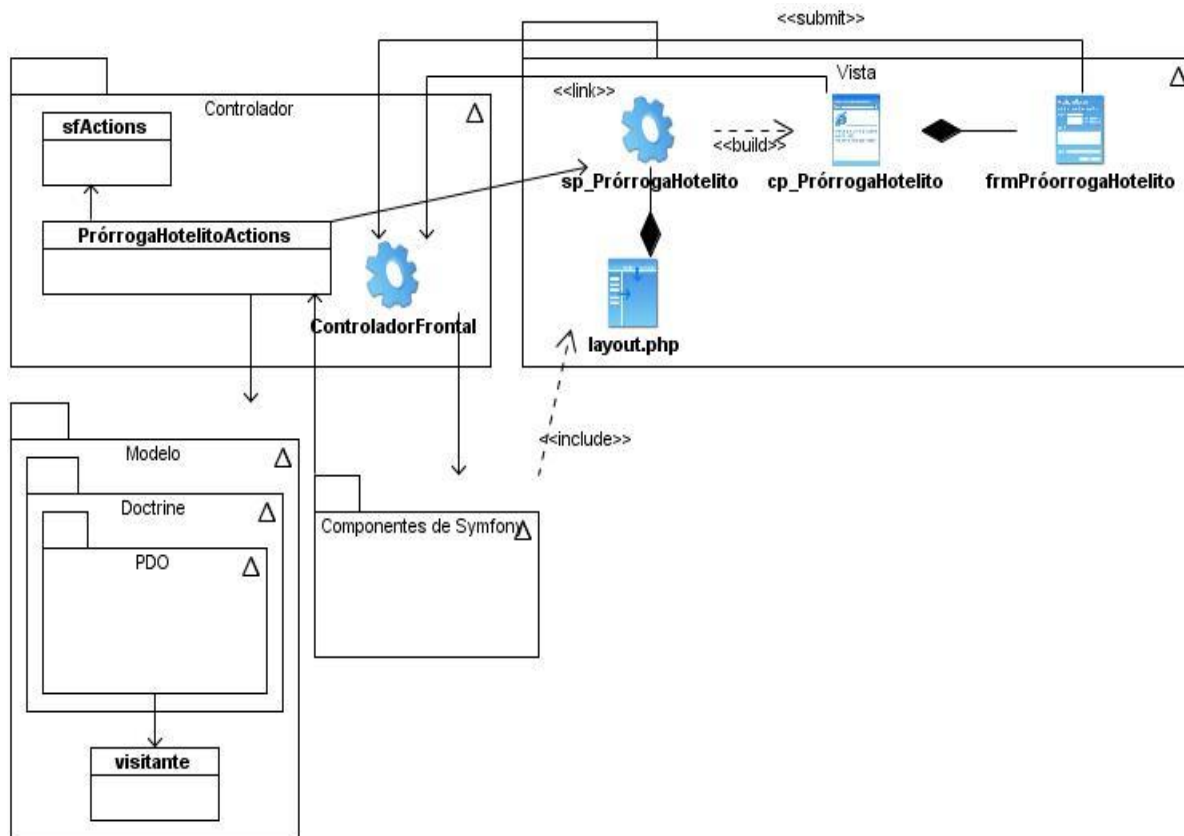


Diagrama de clases del diseño CU Solisitar prórroga para visita al hotelito.

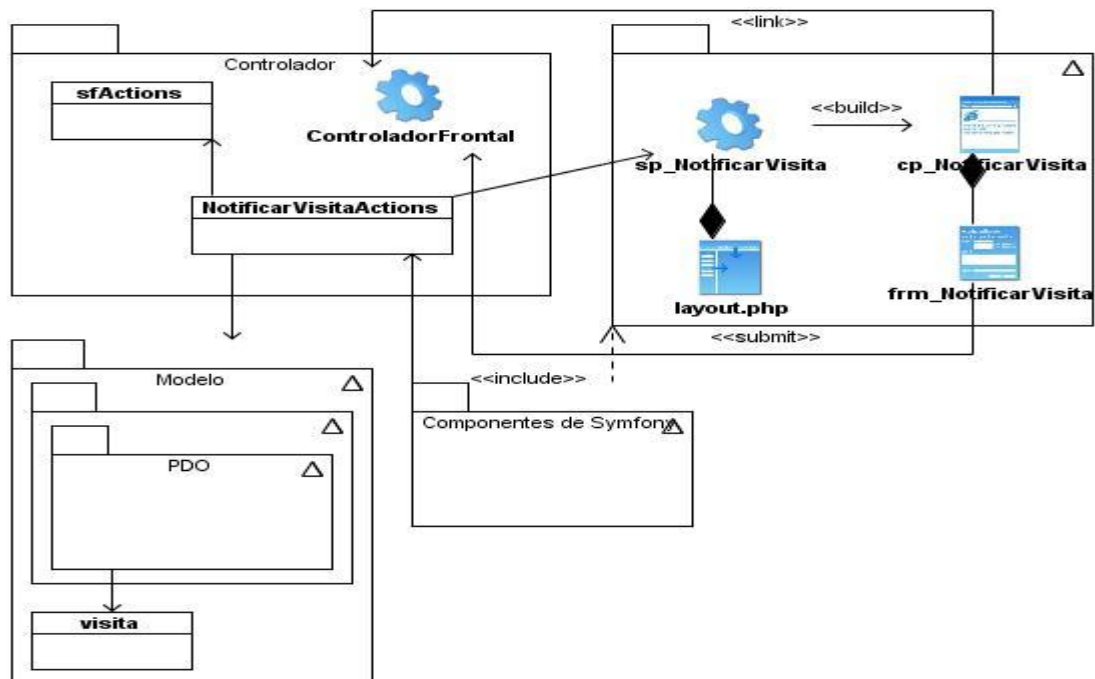


Diagrama de clases del diseño CU Notificar visita

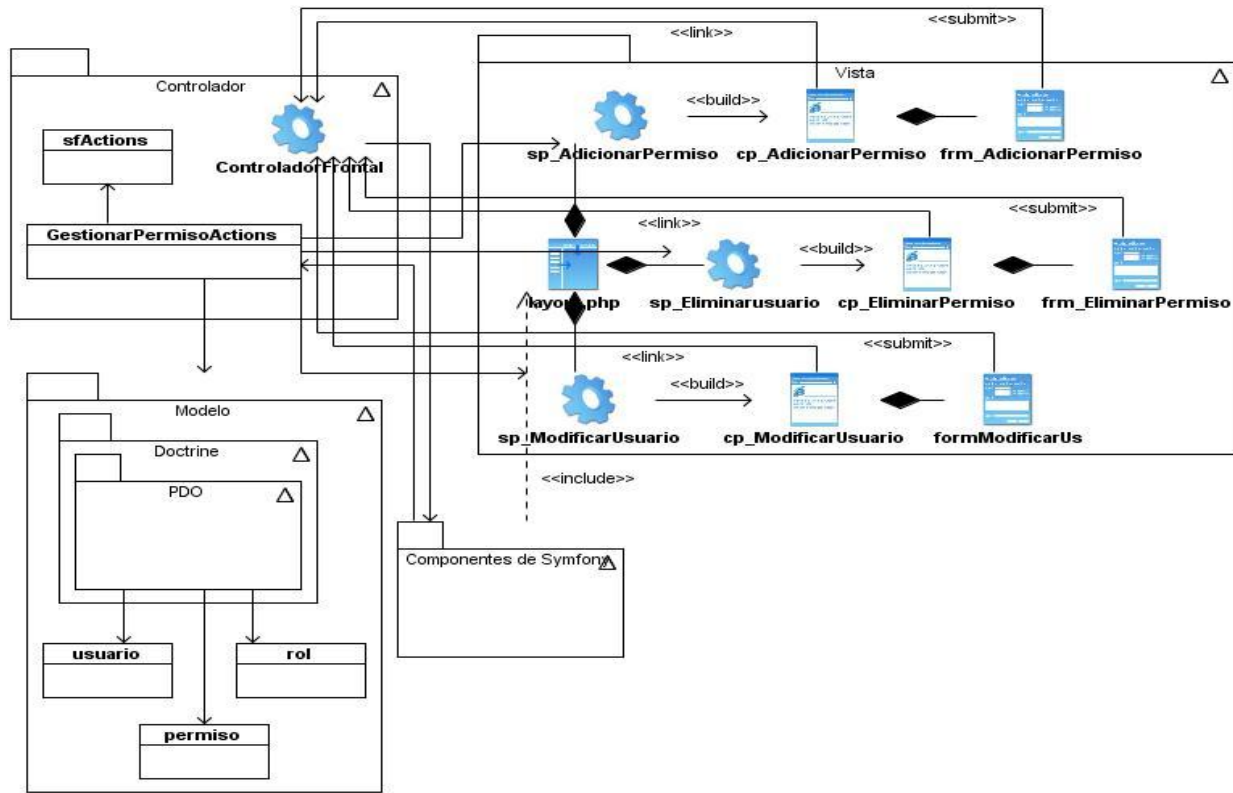


Diagrama de clases del diseño CU gestionar usuario

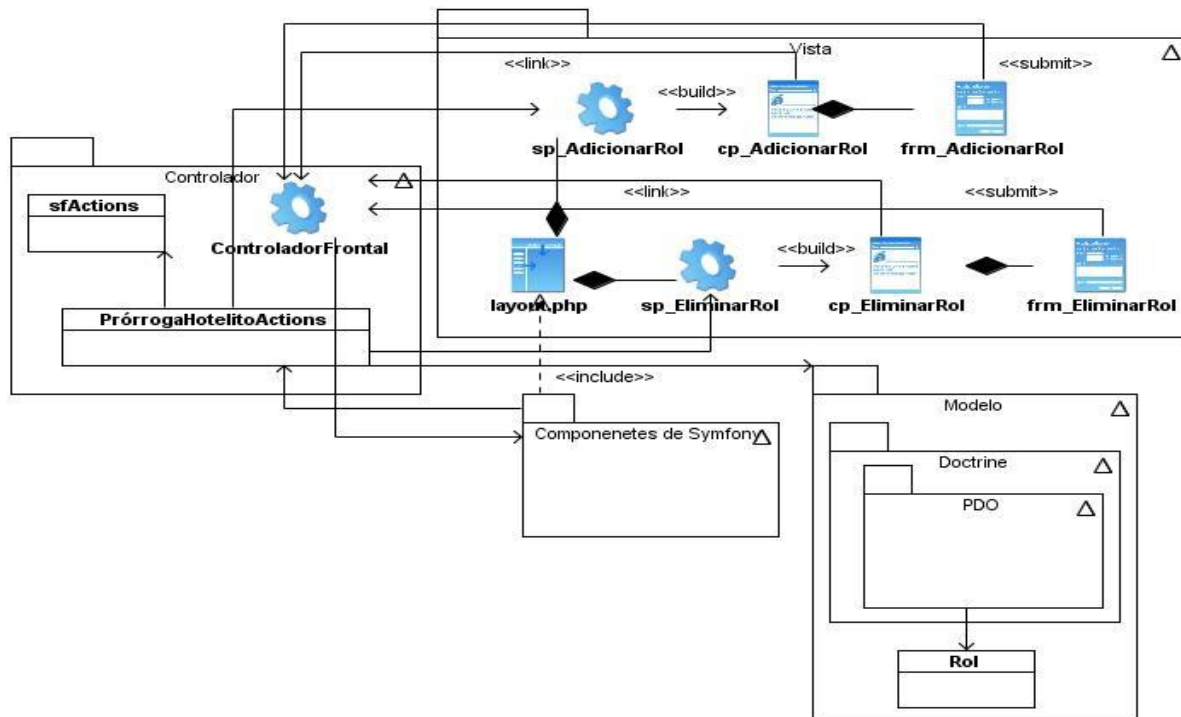


Diagrama de clases del diseño CU Gestionar rol.

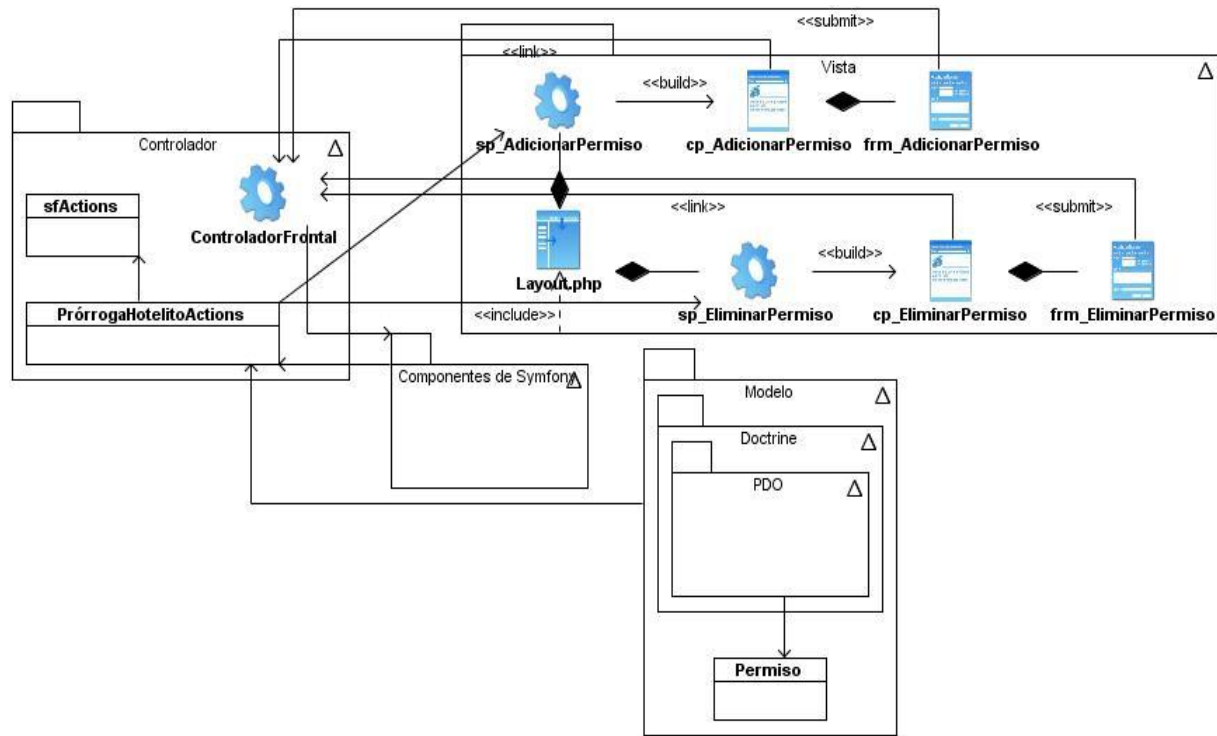


Diagrama de clases del diseño CU Gestionar permiso

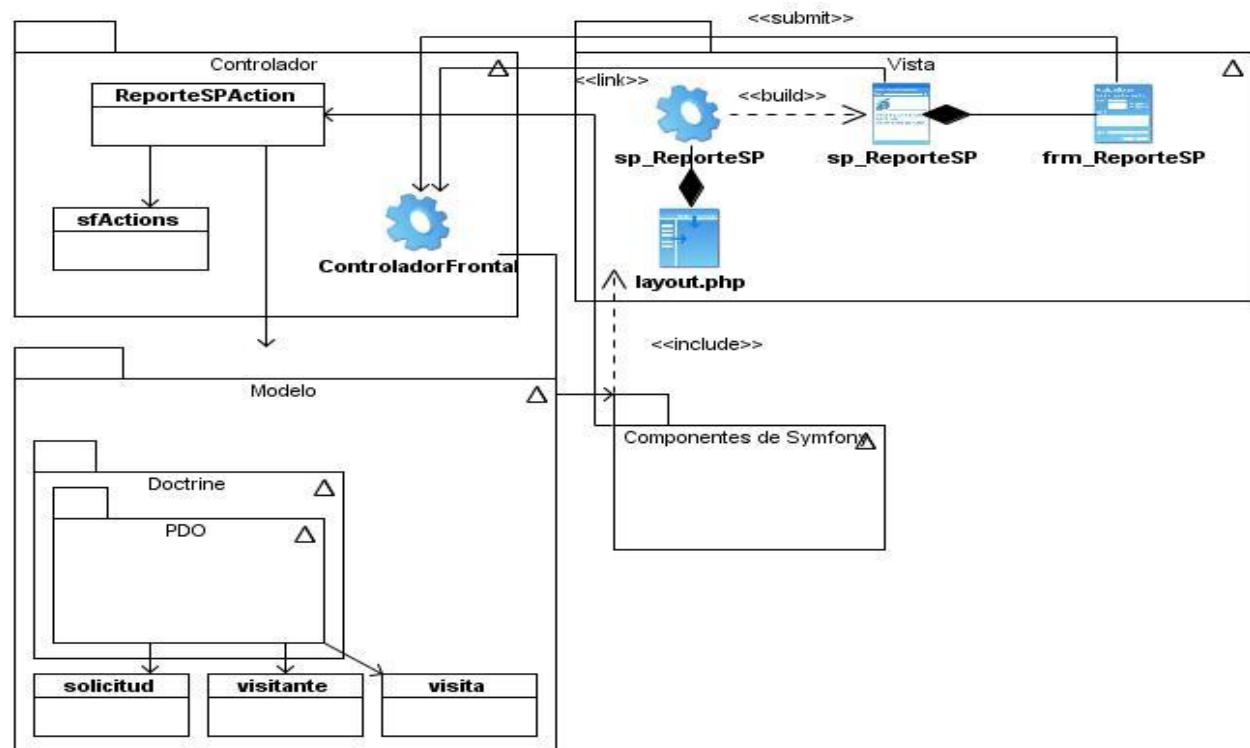


Diagrama de clases del diseño CU Reporte de SP.

Anexo #6 Descripción de las clases del diseño.

Nombre del módulo: solicitud
Nombre de la Clase: solicitudActions.class.php
Nombre del método: executeIndex(sfWebRequest \$request)
Descripción: Muestra todas las solicitudes que ha hecho el usuario que esta autenticado.
Nombre del método: executeAdicionarVisitante(sfWebRequest \$request)
Descripción: Adiciona un nuevo visitante a la solicitud en curso
Nombre del método: executeNuevo(sfWebRequest \$request)
Descripción: Muestra el formulario para adicionar una nueva solicitud
Nombre del método: executeCrear(sfWebRequest \$request)
Descripción: Crea una solicitud
Nombre del método: executeNotificar(sfWebRequest \$request)
Descripción: Crea una solicitud cuya categoría es <i>notificación</i>
Nombre del método: executeEditar(sfWebRequest \$request)
Descripción: Muestra el formulario que permite esditar una solicitud dada
Nombre del método: executeActualiza(sfWebRequest \$request)
Descripción: Actualiza los datos de una solicitud dada
Nombre del método: executeEliminar(sfWebRequest \$request)
Descripción: Elimina una solicitud
Nombre del método: executeMostrar(sfRequest \$request)
Descripción: Muestra los detalles de una solicitud
Nombre del método: executeEvaluar(sfRequest \$request)
Descripción: Muestra el formulario que permite aprobar o denegar una solicitud
Nombre del método: executeAprobar(sfRequest \$request)
Descripción: Aprueba una solicitud
Nombre del método: executeDenegar(sfRequest \$request)
Descripción:Deniega una solicitud
Nombre del método: executeHistorial(sfWebRequest \$request)
Descripción: Muestra el hostorial de usuario autenticado
Nombre del método: executeListarRecibidas(sfWebRequest \$request)
Descripción: Lista todas las solicitudes hechas a un directivo determinado que éste no ha revisado. (Las solicitudes que están en estado pendiente)
Nombre del método: executeListarEnviadas(sfWebRequest \$request)

Descripción: Lista las solicitudes hechas por un usuario que no han sido revisadas por el directivo a quien se le envió (Las solicitudes que están en estado pendiente)
Nombre del método: executeListarUltimasEnviadas(sfWebRequest \$request)
Descripción: Lista las solicitudes hechas por un usuario que ya fueron revisadas
Nombre del método: executeListarUltimasRevisadas(sfWebRequest \$request)
Descripción: Lista las solicitudes hechas por un usuario que ya fueron revisadas

Nombre del módulo: visita
Nombre de la Clase: visitaActions.class.php
Nombre del método: executeIniciarProrrogar(sfWebRequest \$request)
Descripción: Lista las visitas en curso que pueden ser prorogadas
Nombre del método: executeProrrogar(sfWebRequest \$request)
Descripción: Prorroga una visita determinada

Nombre del módulo: visitante
Nombre de la Clase: visitanteActions.class.php
Nombre del método: executeListar(sfWebRequest \$request)
Descripción: Descripción: Lista los visitantes
Nombre del método: executeMostrar(sfWebRequest \$request)
Descripción: Descripción: Muestra los detalles de un visitante
Nombre del método: executeNuevo(sfWebRequest \$request)
Descripción: Muestra el formulario para adicionar un nuevo visitante
Nombre del método: executeCrear(sfWebRequest \$request)
Descripción: Crea un nuevo visitante
Nombre del método: executeEditar(sfWebRequest \$request)
Descripción: Muestra el formulario para editar un visitante dado
Nombre del método: executeActualizar(sfWebRequest \$request)
Descripción: Actualiza los datos de un visitante
Nombre del método: executeSeleccionar(sfWebRequest \$request)
Descripción: Selecciona de una lista un visitante determinado
Nombre del método: executeBuscarPorSolicitante(sfWebRequest \$request)
Descripción: Lista todos los visitantes que han estado en alguna solicitud de un solicitante dado

Nombre del método: executeEliminar(sfWebRequest \$request)
Descripción: Elimina un visitante dado
Nombre del método: executeBuscarVisitanteEnCursoCi(sfWebRequest \$request)
Descripción: Lista los visitantes en curso según el CI
Nombre del método: executeBuscarVisitantePendienteCi(sfWebRequest \$request)
Descripción: Lista los visitantes pendientes según el CI
Nombre del método: executeRegistrarSalida(sfWebRequest \$request)
Descripción: Registra la salida de un visitante
Nombre del método: executeRegistrarEntrada(sfWebRequest \$request)
Descripción: Registra la entrada de un visitante

Nombre del módulo: area
Nombre de la clase: areaActions.class.php
Nombre del método: executeDirectivosPorArea(sfWebRequest \$request)
Descripción: Muestro los directivos de un área determinada

Anexo #7 Diagramas de secuencia.

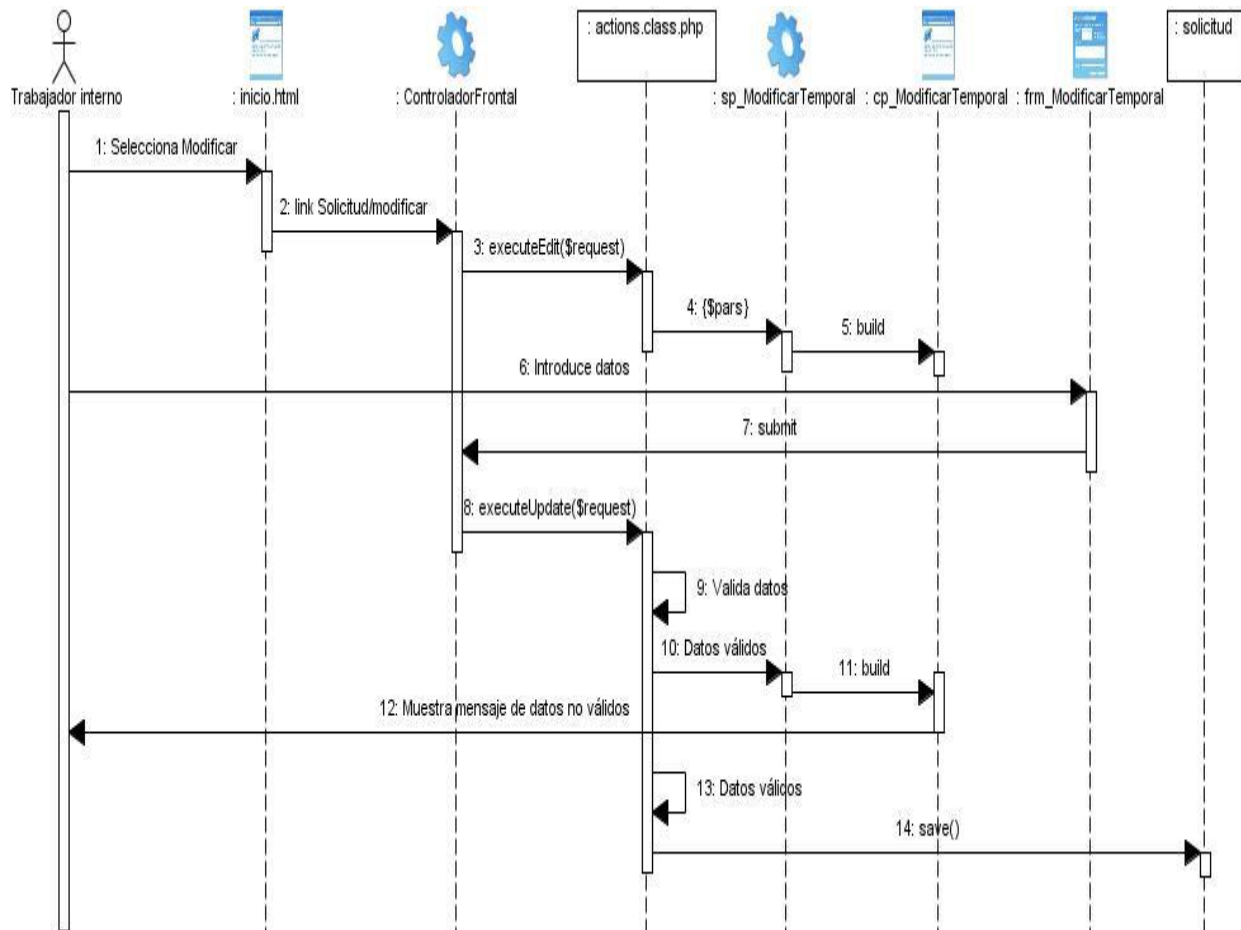


Diagrama de secuencia CU Gestionar visita temporal. Sección Modificar visita temporal.

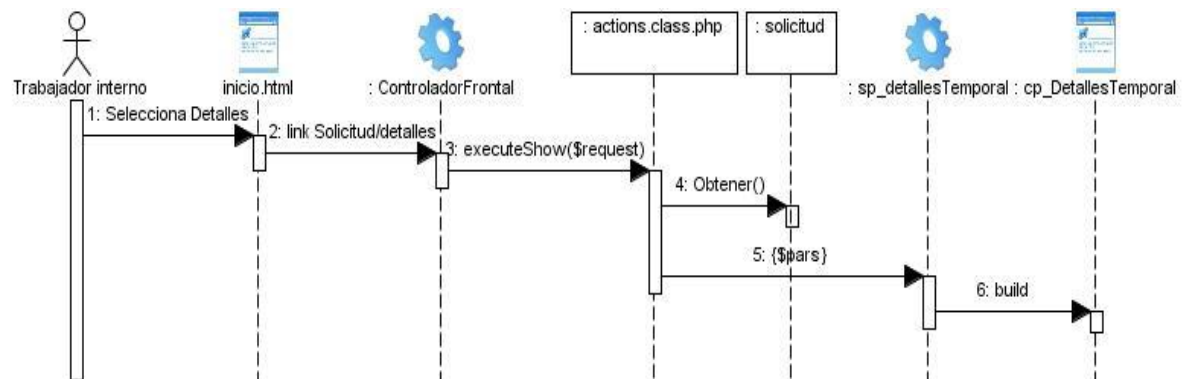


Diagrama de secuencia CU Gestionar visita temporal. Sección Detalles visita temporal.

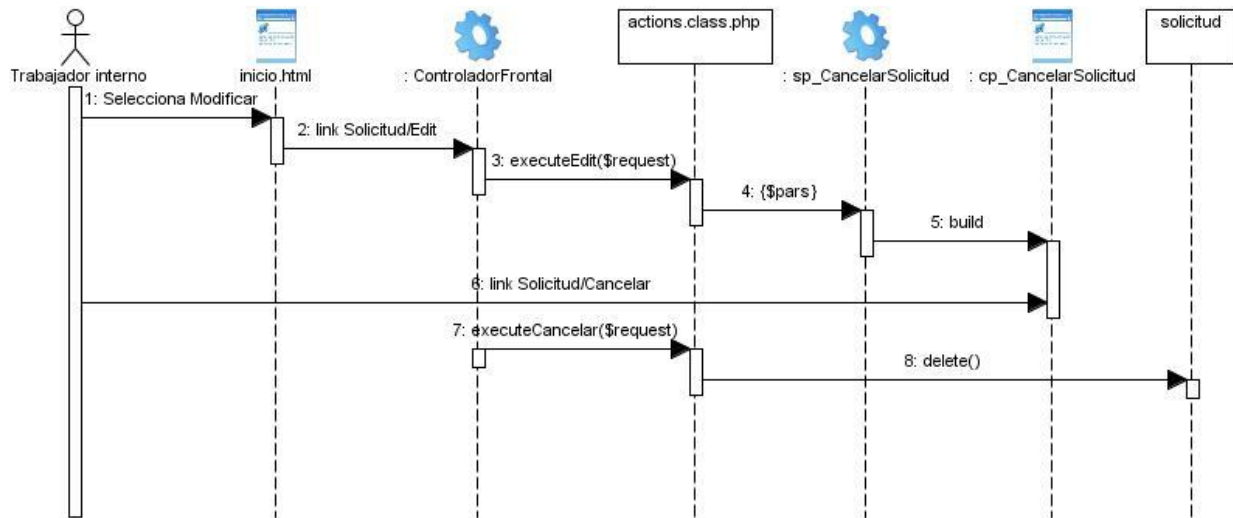


Diagrama de secuencia CU Gestionar visita temporal. Sección Cancelar visita temporal.

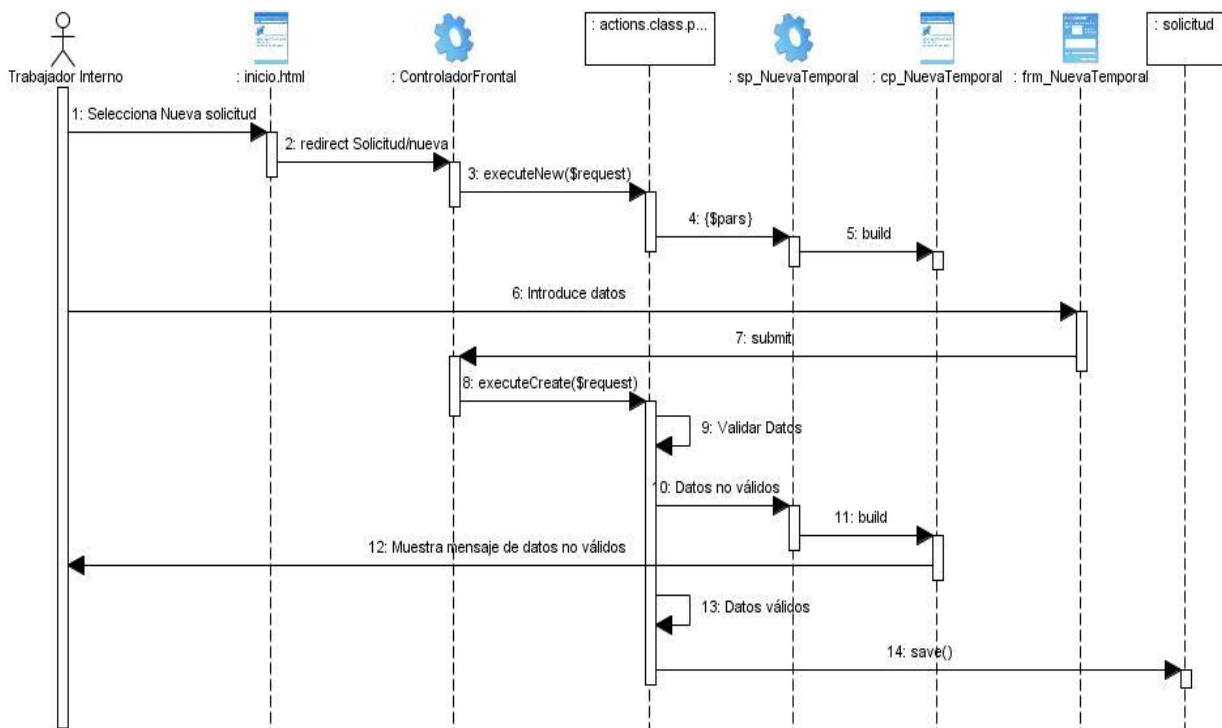


Diagrama de secuencia CU Gestionar visita temporal. Sección Nueva visita temporal.

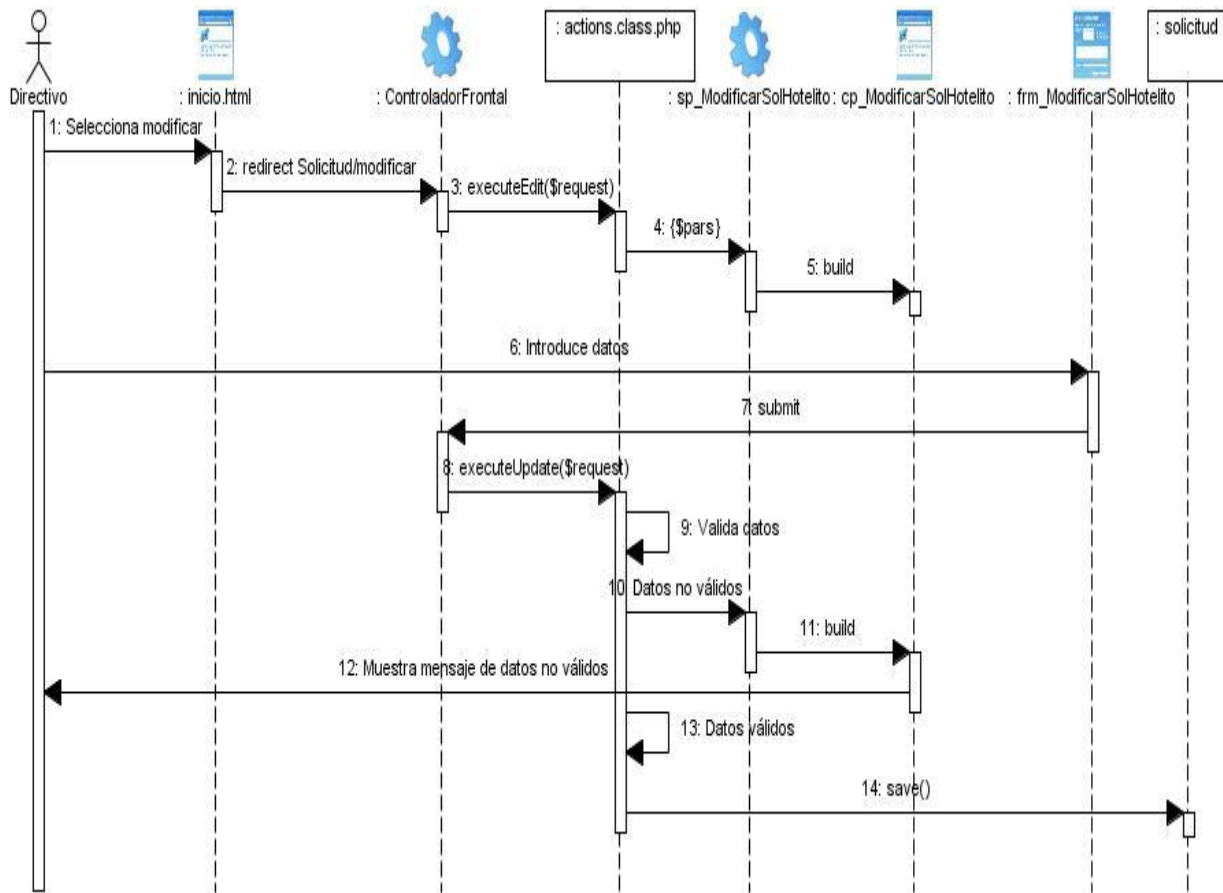


Diagrama de secuencia CU Gestionar visita al hotelito. Sección Modificar visita al hotelito.

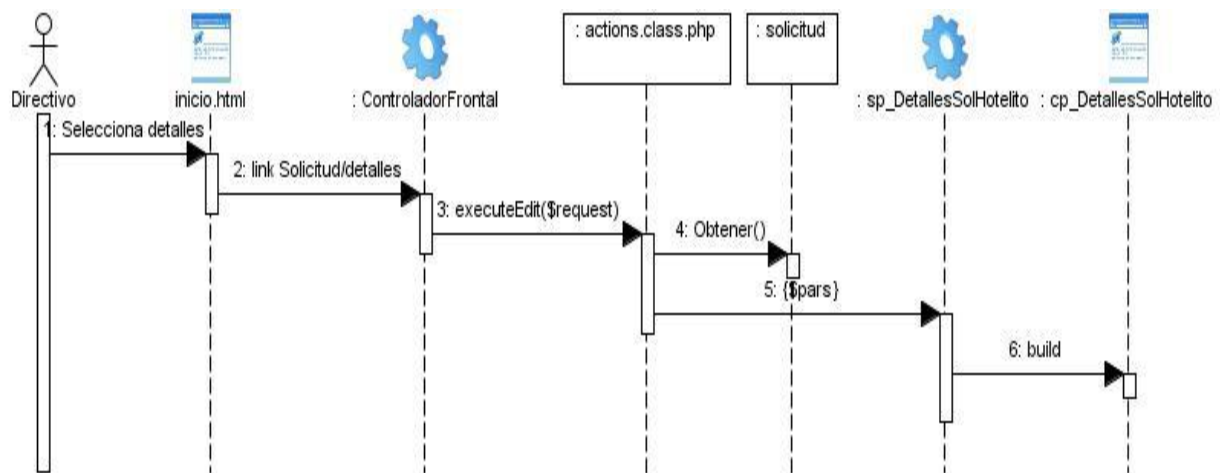


Diagrama de secuencia CU Gestionar visita al hotelito. Sección Detalles visita al hotelito.

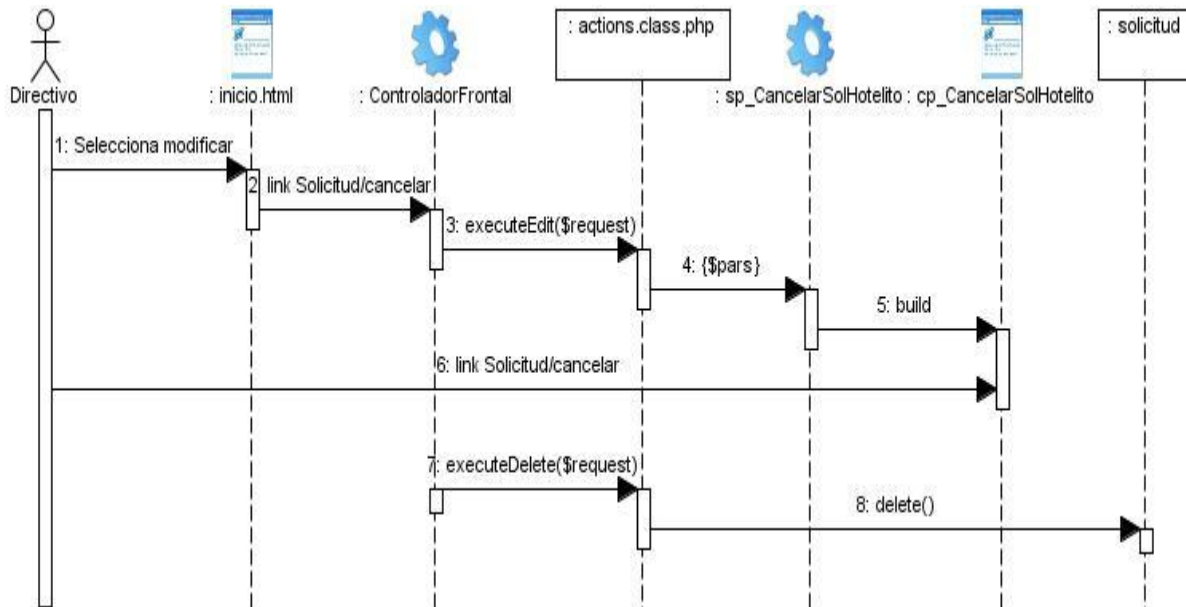


Diagrama de secuencia CU Gestionar visita al hotelito. Sección Cancelar visita al hotelito.

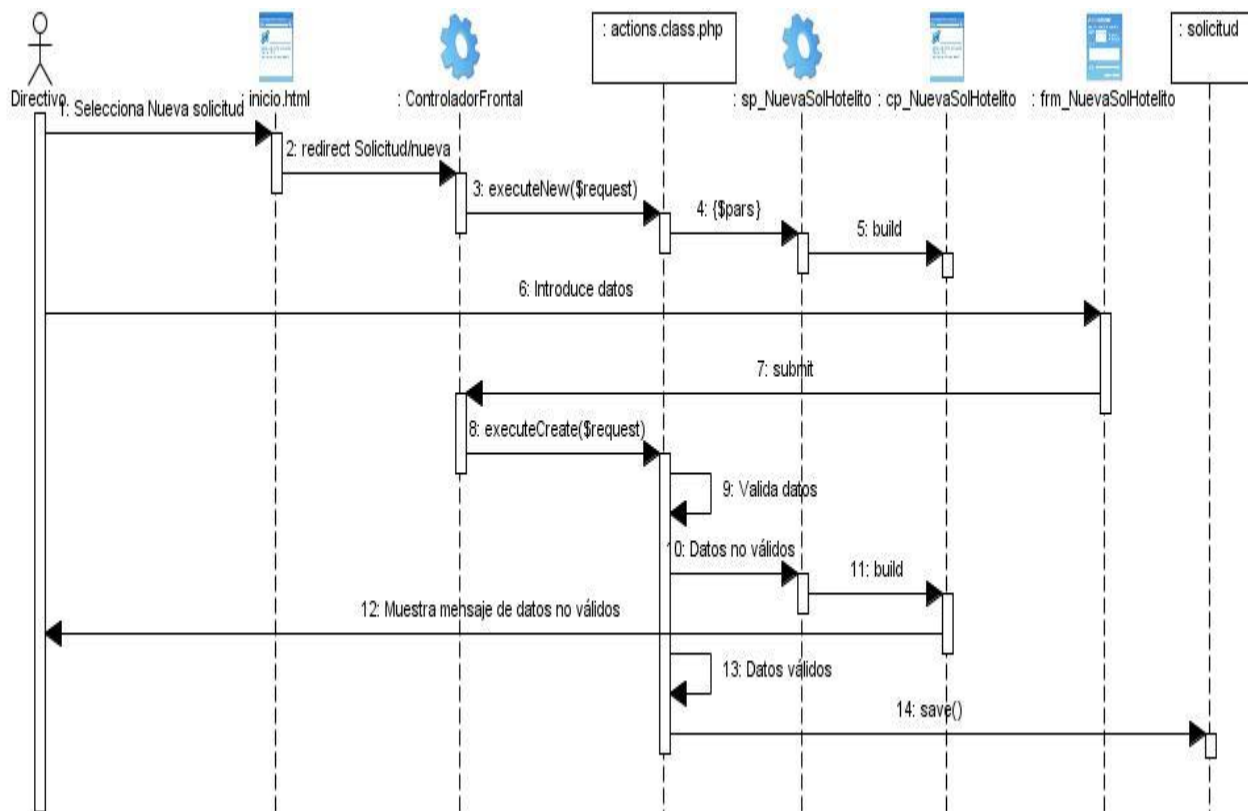


Diagrama de secuencia CU Gestionar visita al hotelito. Sección Nueva visita al hotelito.

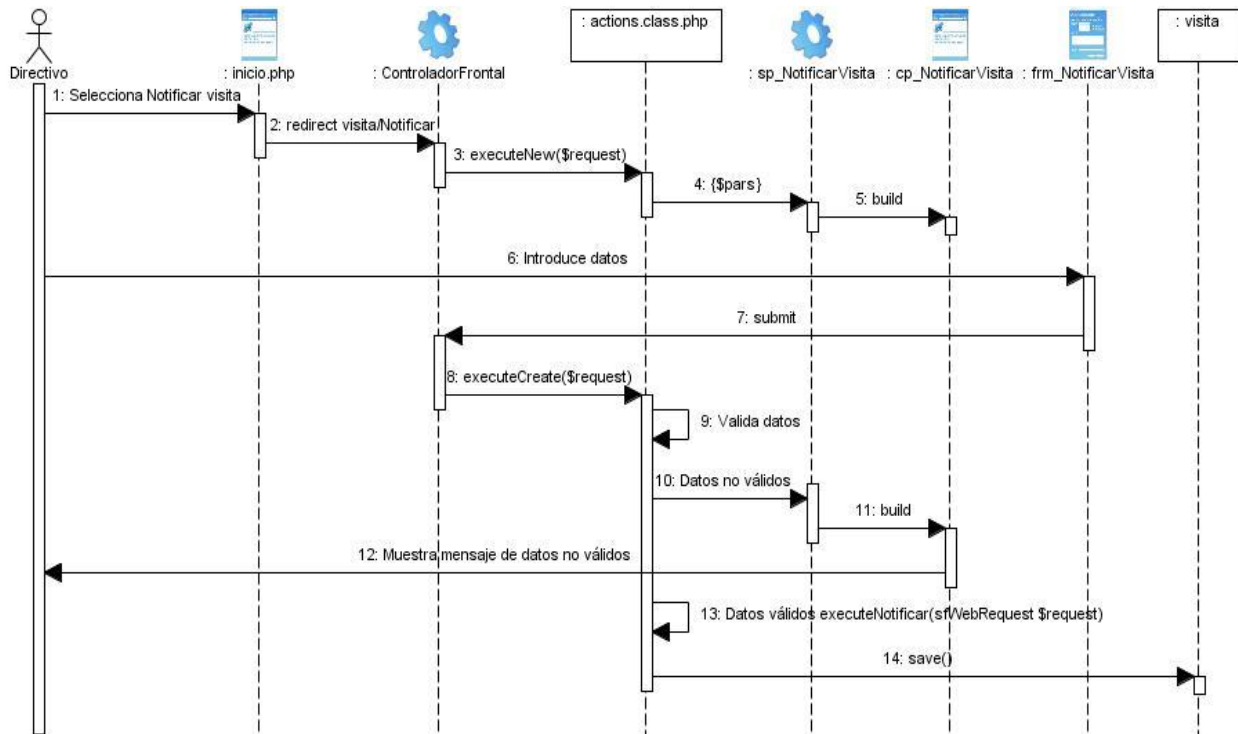


Diagrama de secuencia CU Notificar visita.

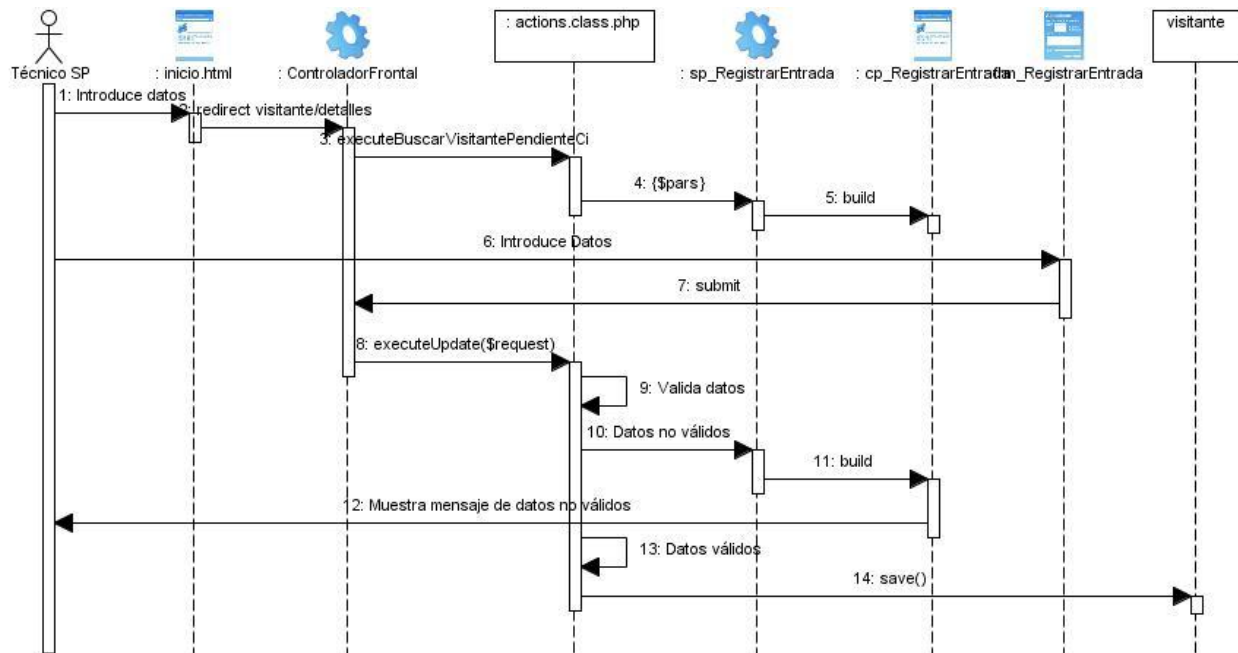


Diagrama de secuencia CU Registrar entrada/salida. Sección Registrar entrada.

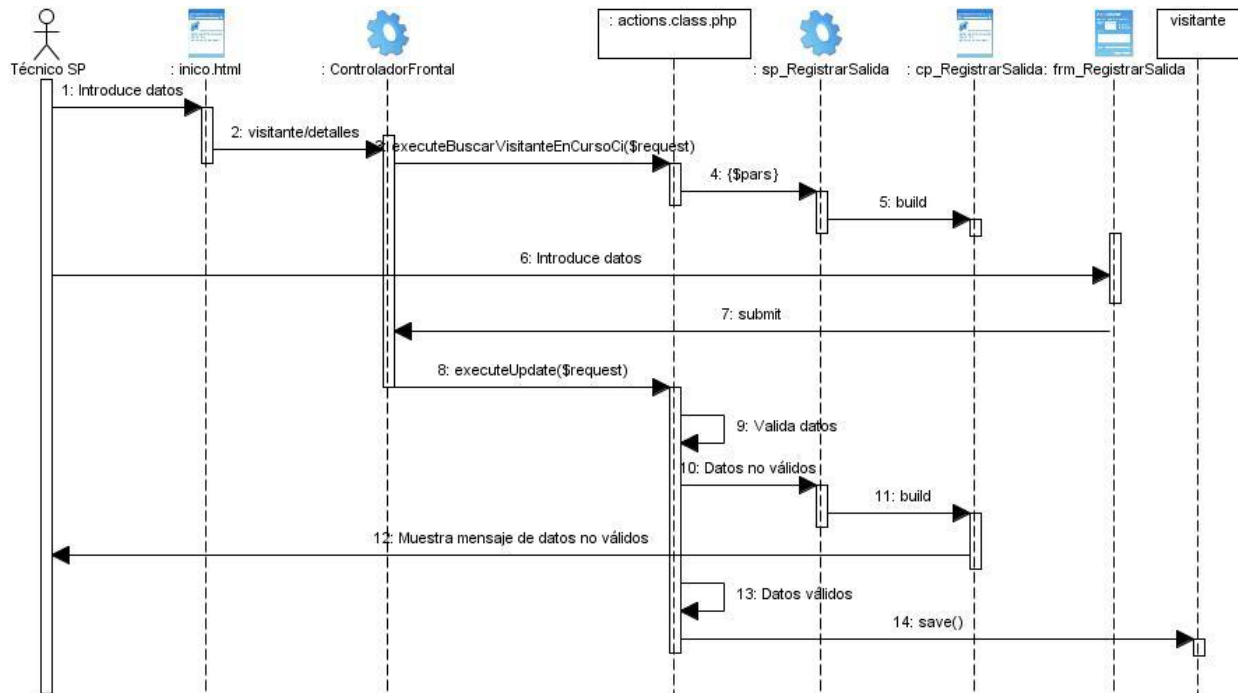


Diagrama de secuencia CU Registrar entrada/salida. Sección Registrar salida.

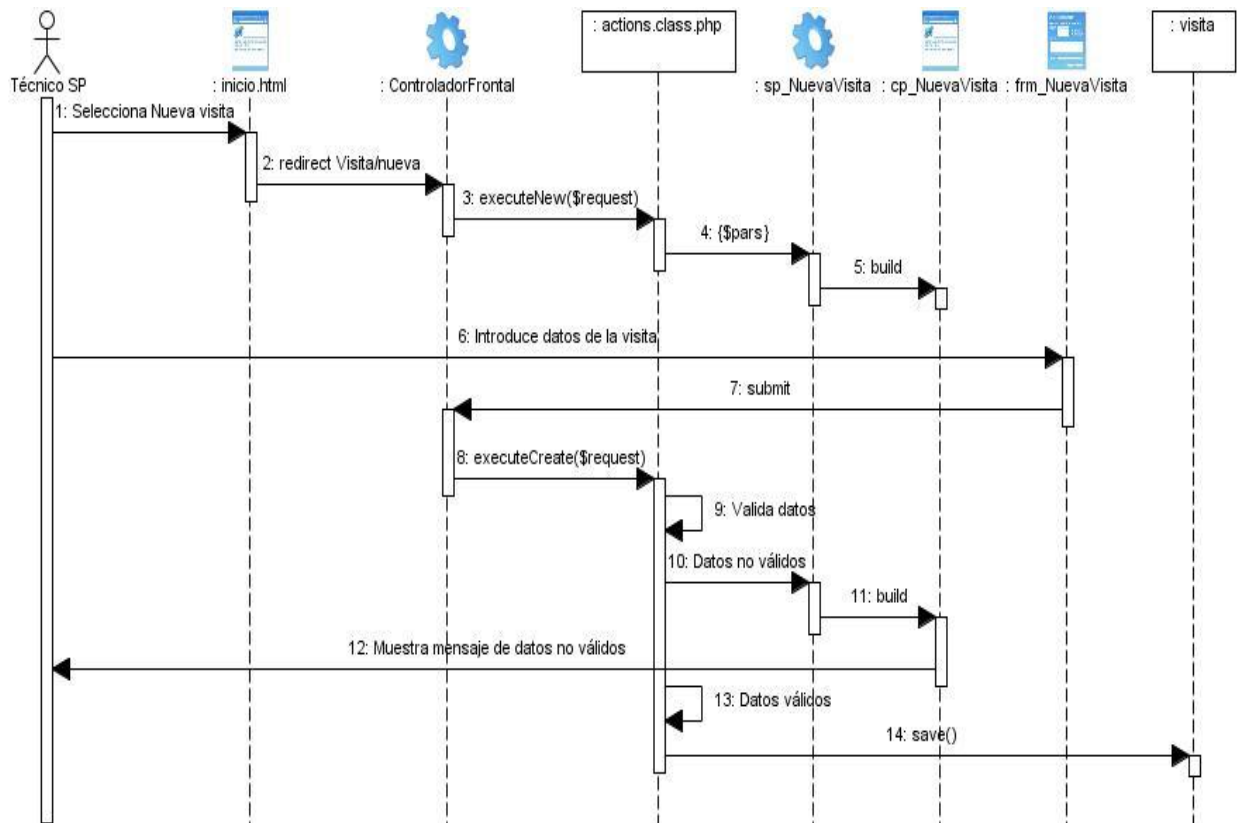


Diagrama de secuencia CU Registrar entrada/salida. Sección Registrar nueva visita.

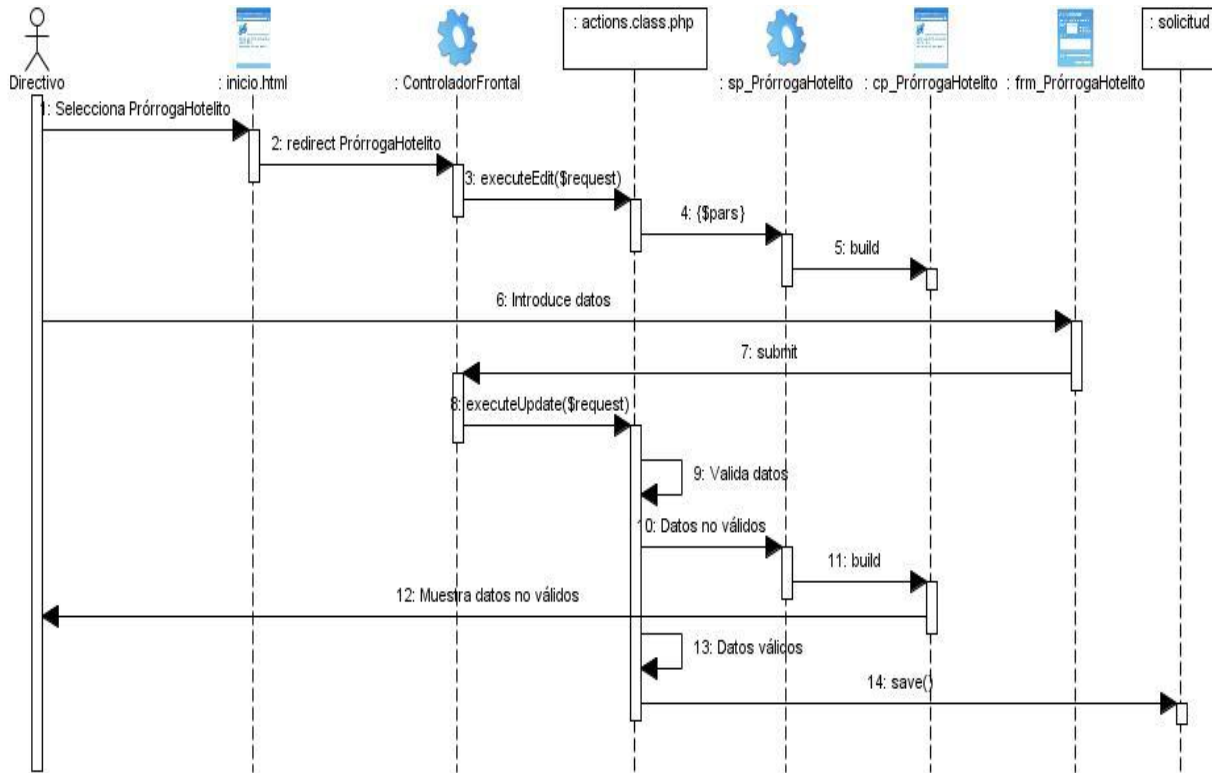


Diagrama de secuencia CU Registrar Prósroga hotelito.

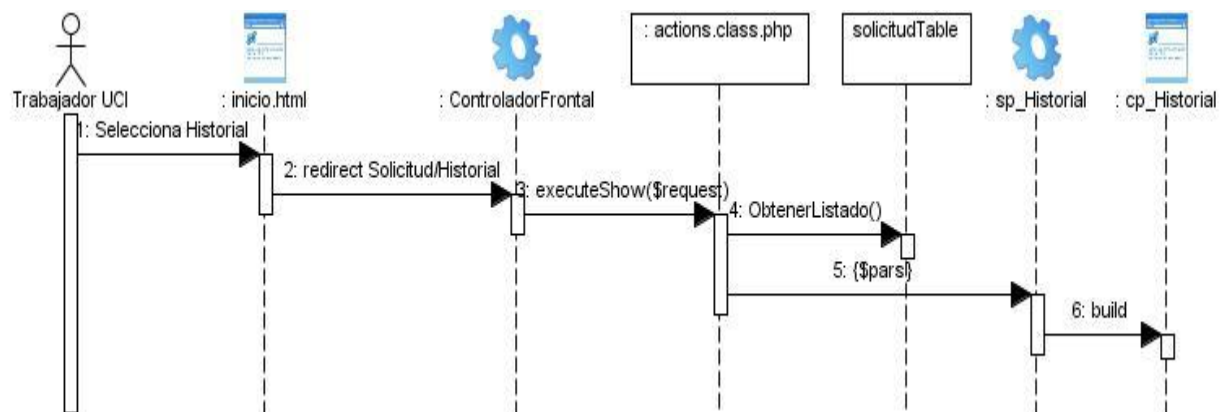


Diagrama de secuencia CU Mostrar historial.

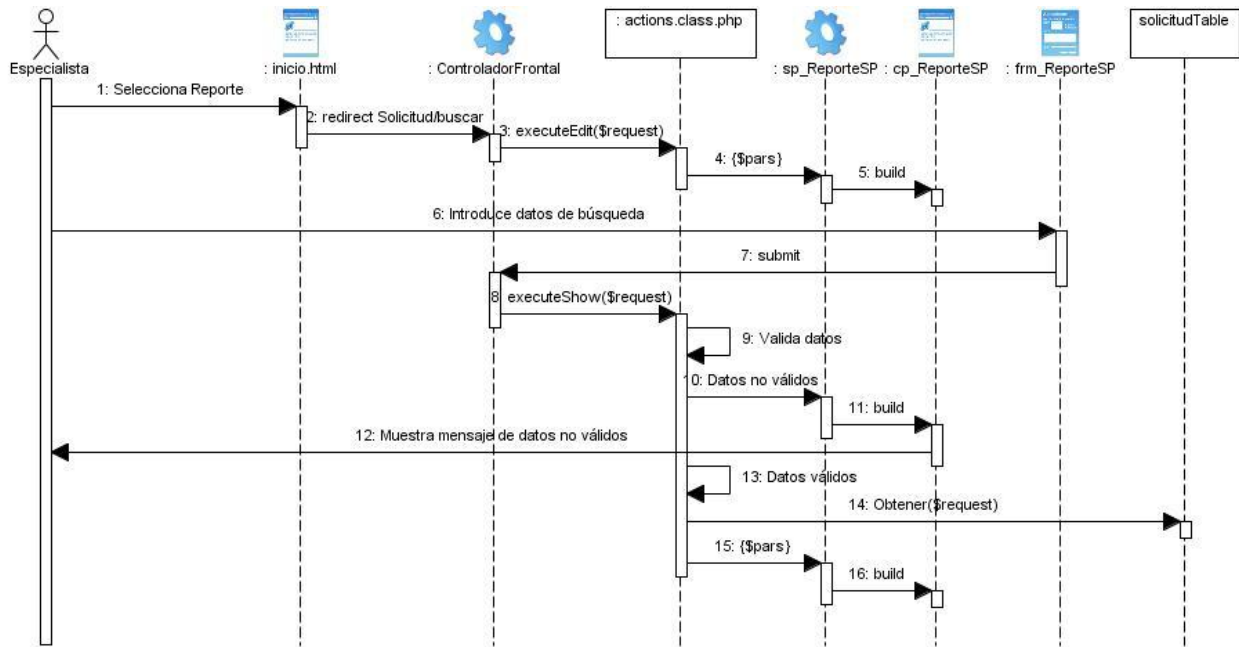


Diagrama de secuencia CU Reporte de SP.

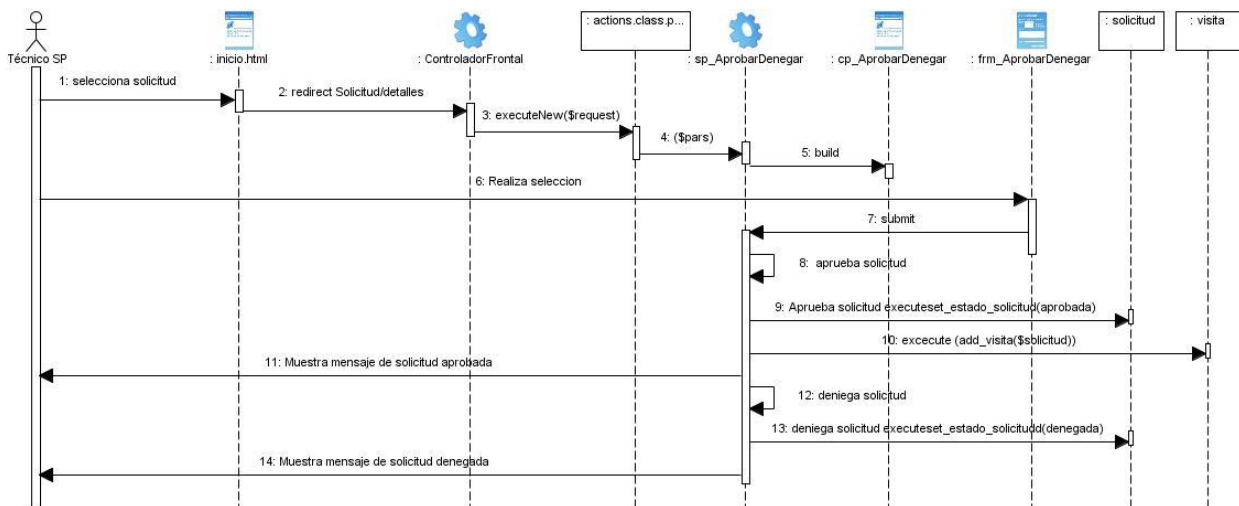


Diagrama de secuencia CU Aprobar/denegar.

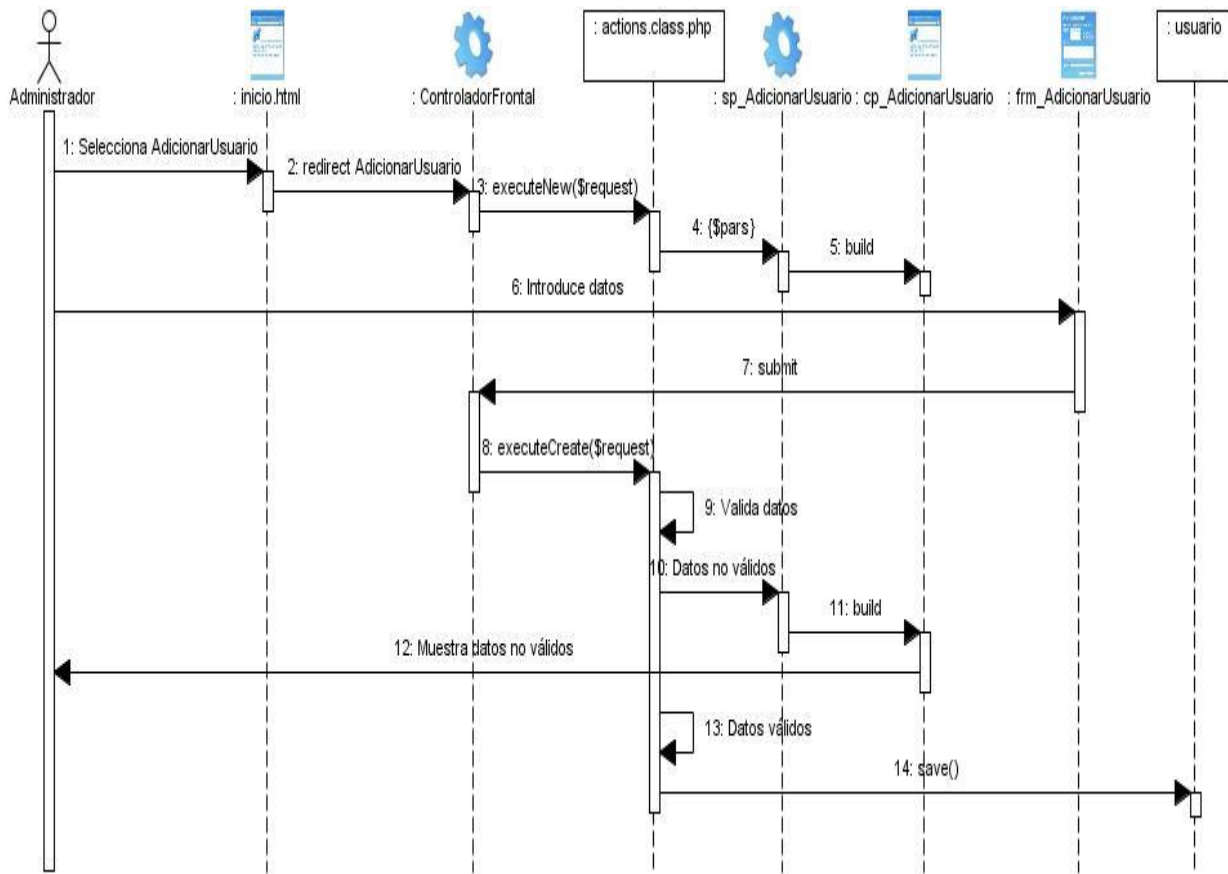


Diagrama de secuencia CU Gestionar Usuario. Sección Nuevo usuario.

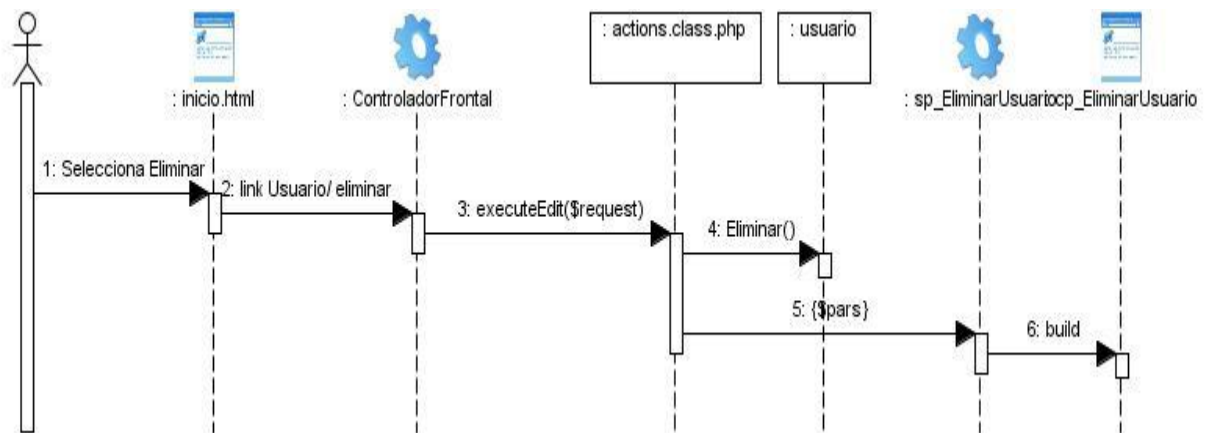


Diagrama de secuencia CU Gestionar Usuario. Sección Eliminar usuario

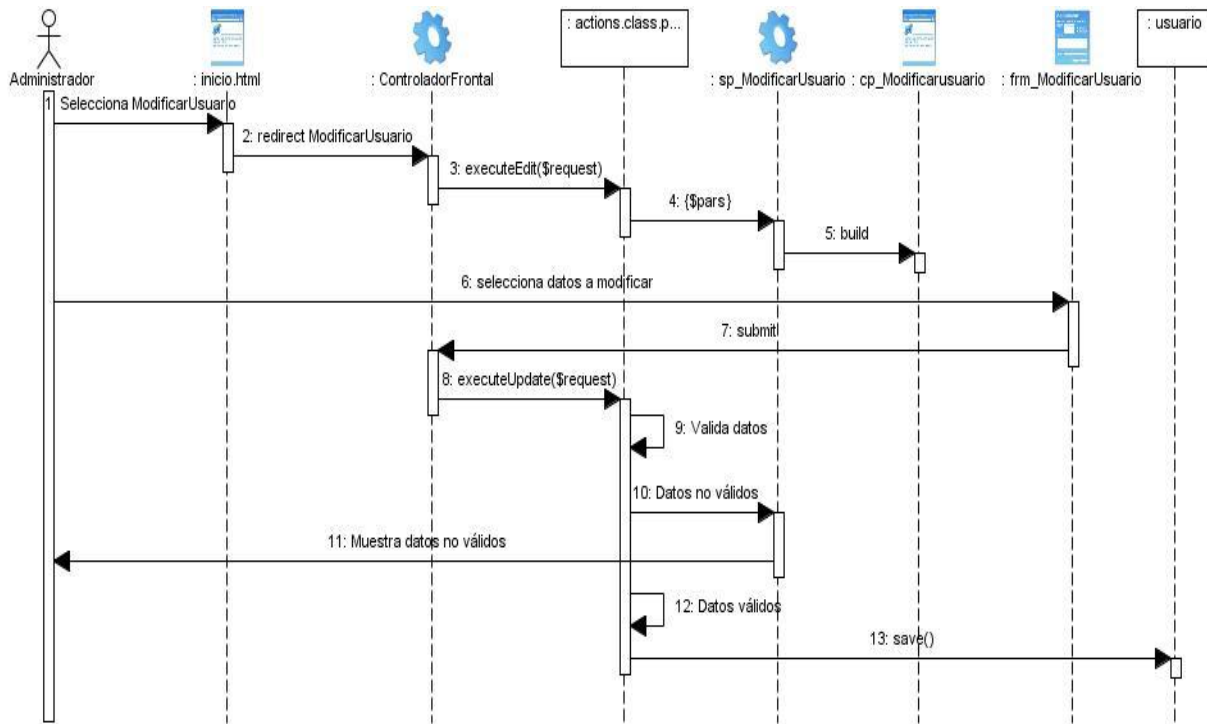


Diagrama de secuencia CU Gestionar Usuario. Sección Modificar usuario.

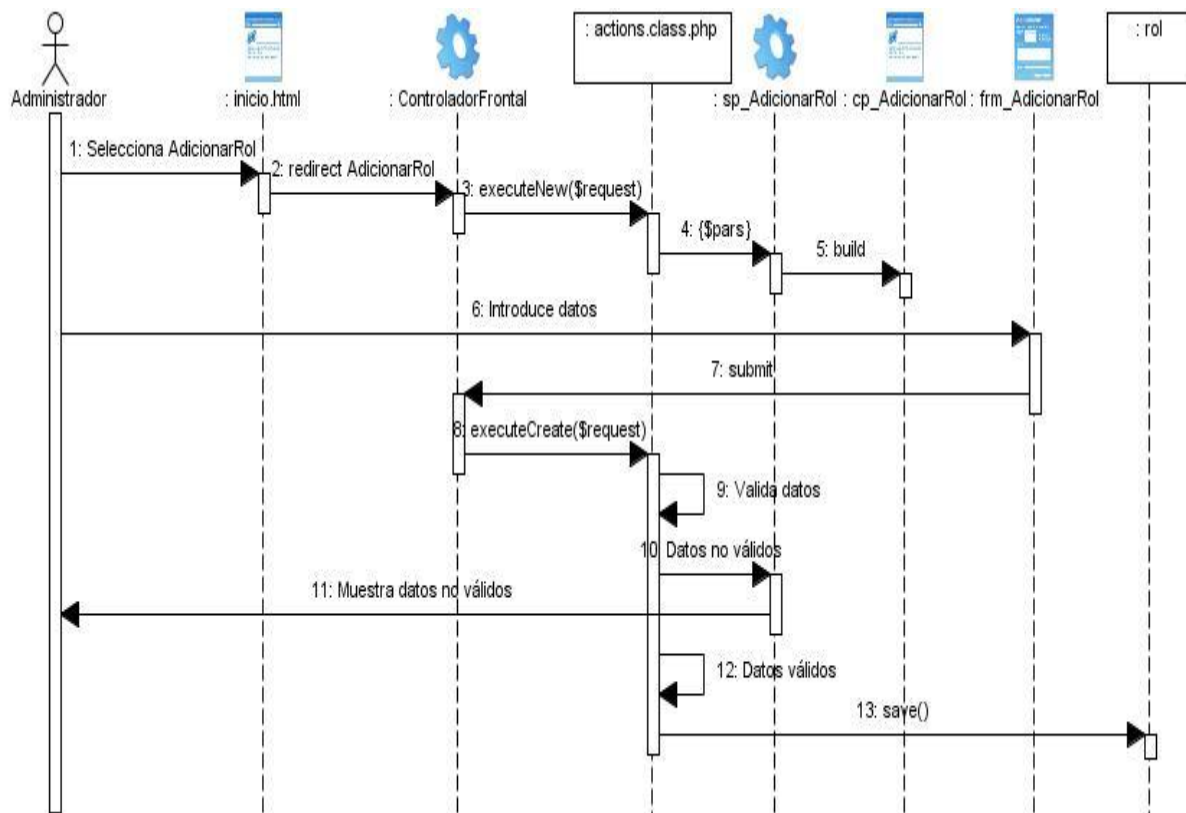


Diagrama de secuencia CU Gestionar rol. Sección Adicionar rol.

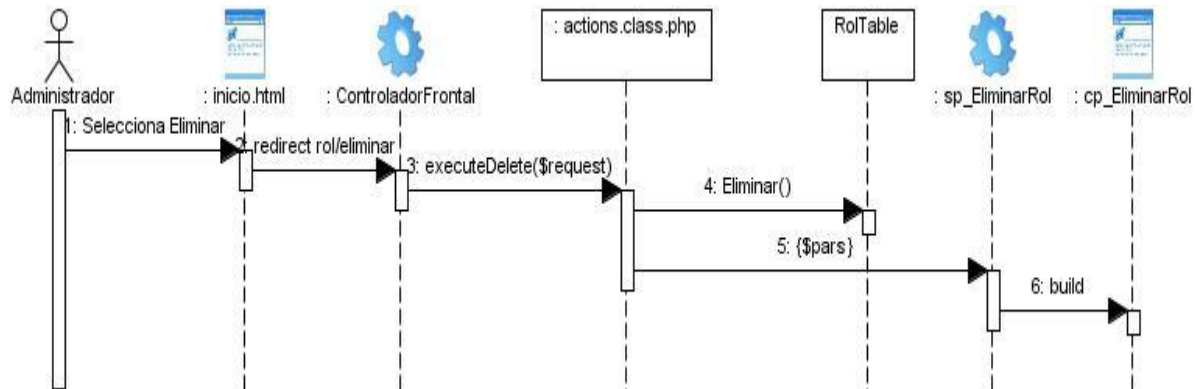


Diagrama de secuencia CU Gestionar Rol. Sección Eliminar rol.

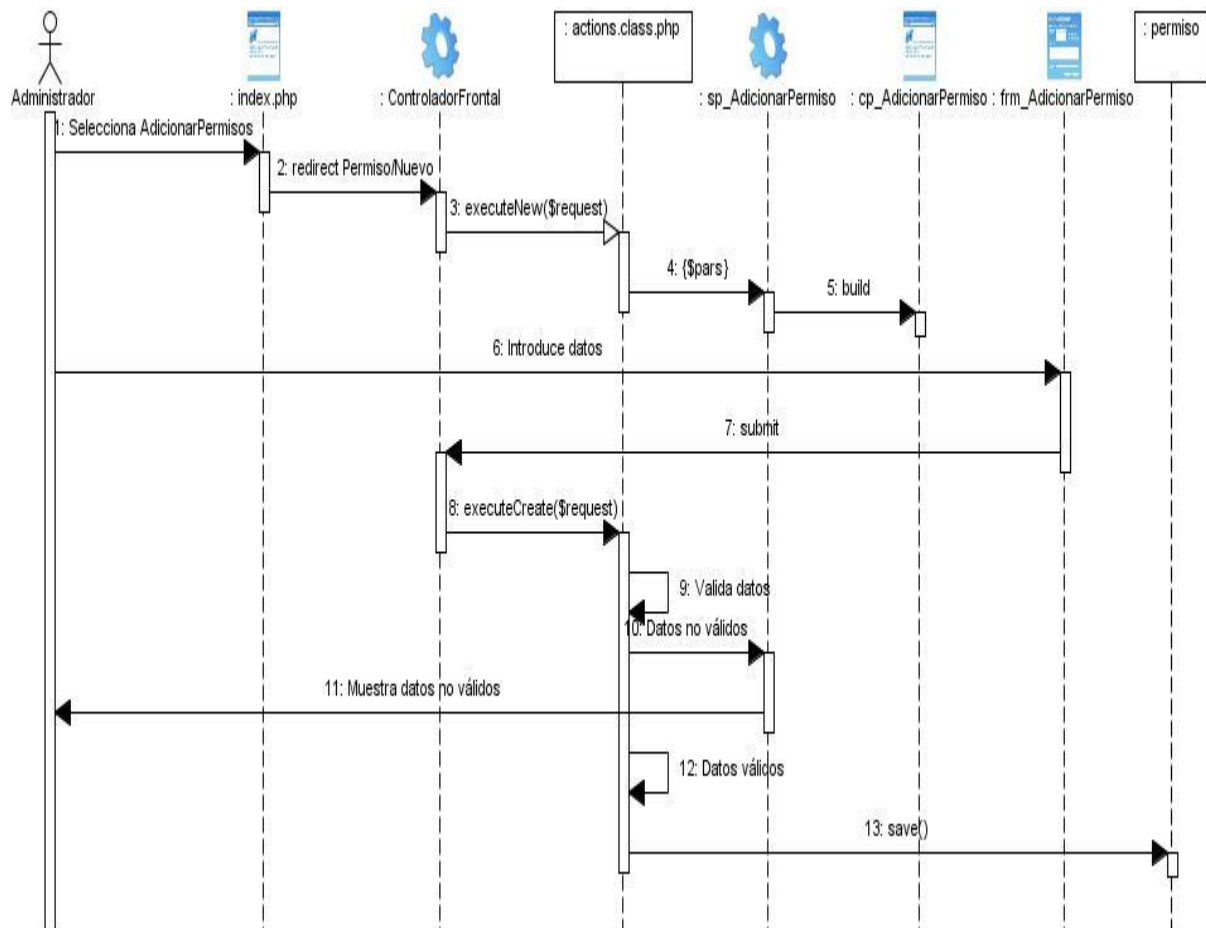


Diagrama de secuencia CU Gestionar permiso. Sección Adicionar Permiso.

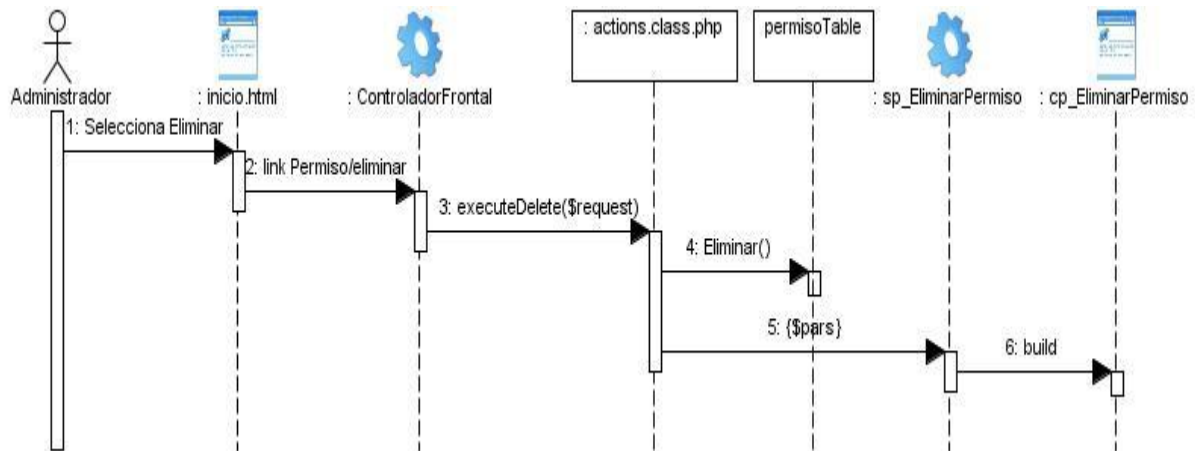


Diagrama de secuencia CU Gestionar Permiso. Sección Eliminar permiso.

Anexo # 8 Diagramas de componentes.

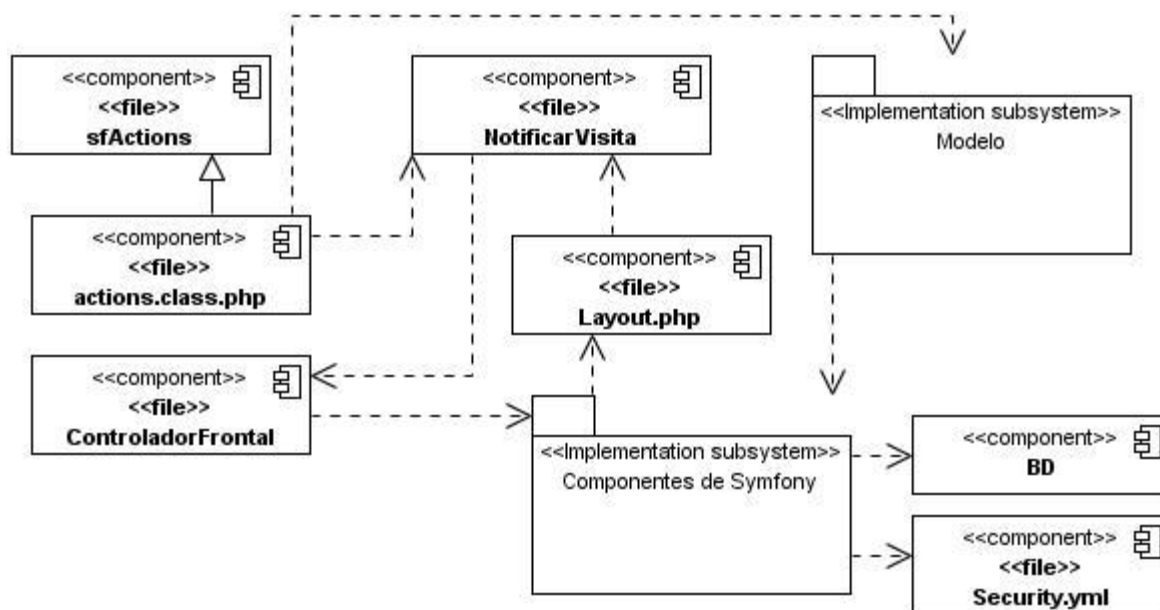


Diagrama de componentes CU Notificar visita.

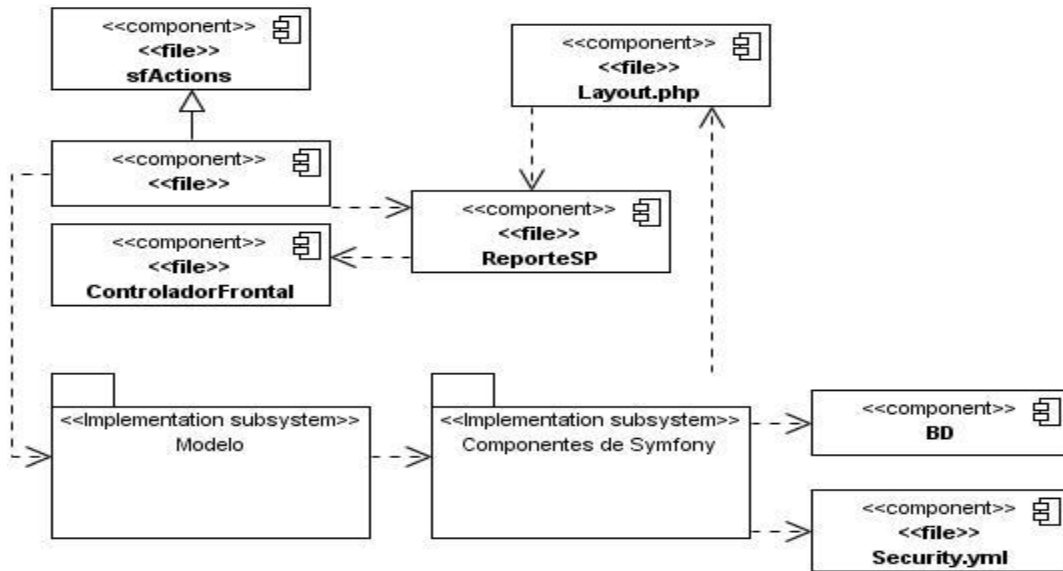


Diagrama de componentes CU Reporte de SP.

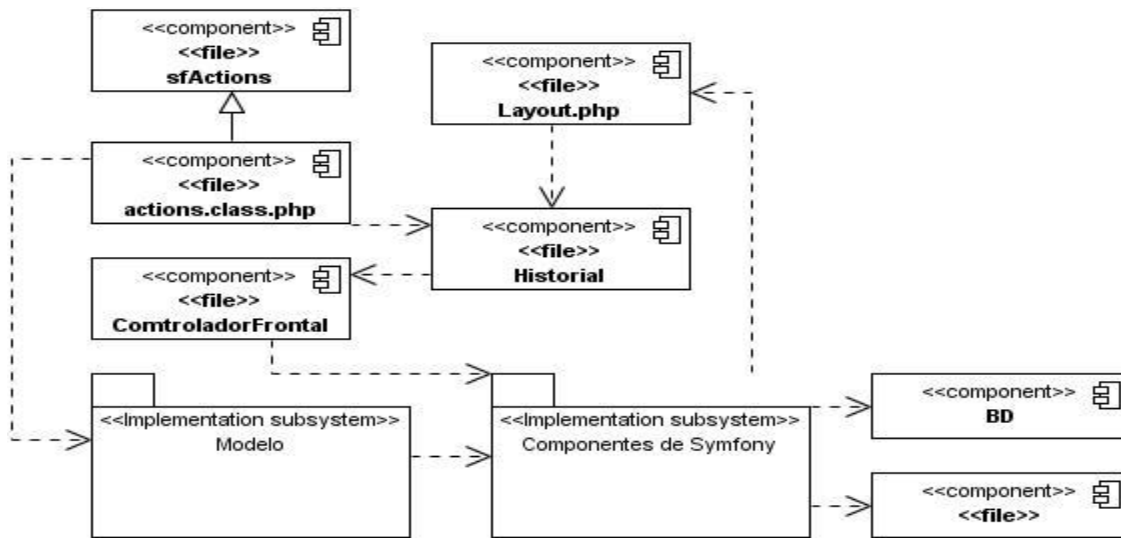


Diagrama de componentes CU Mostrar historial.

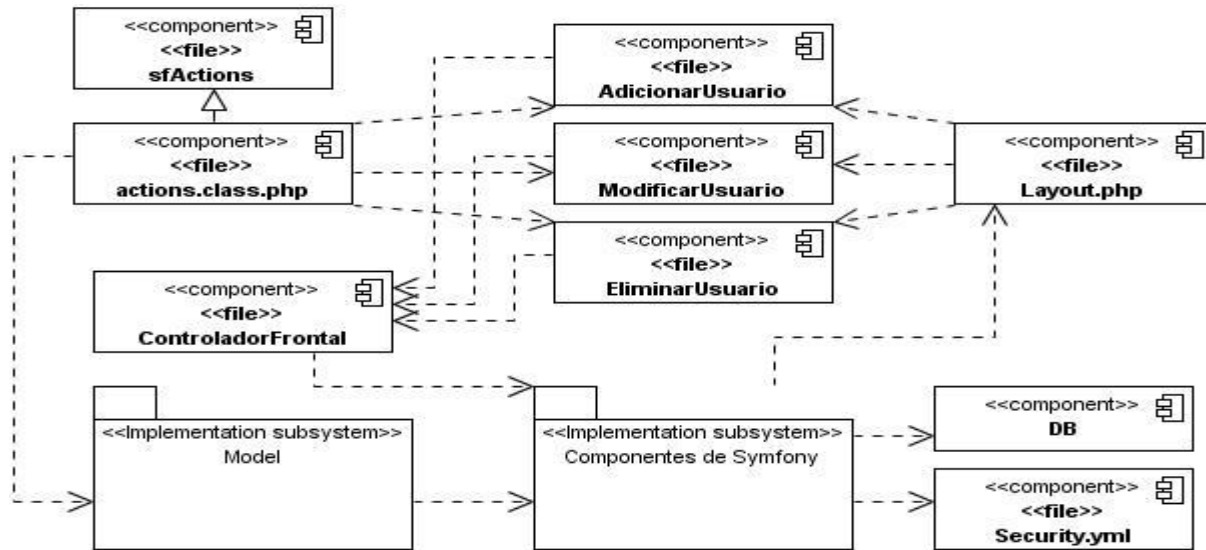


Diagrama de componentes CU Gestionar usuario.

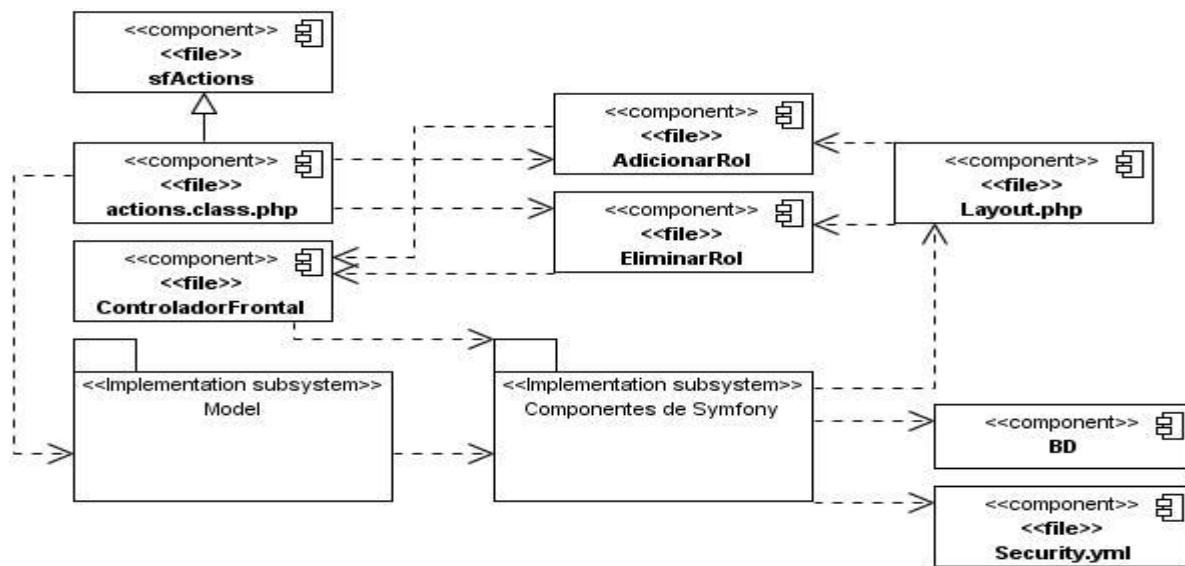


Diagrama de componentes CU Gestionar rol.

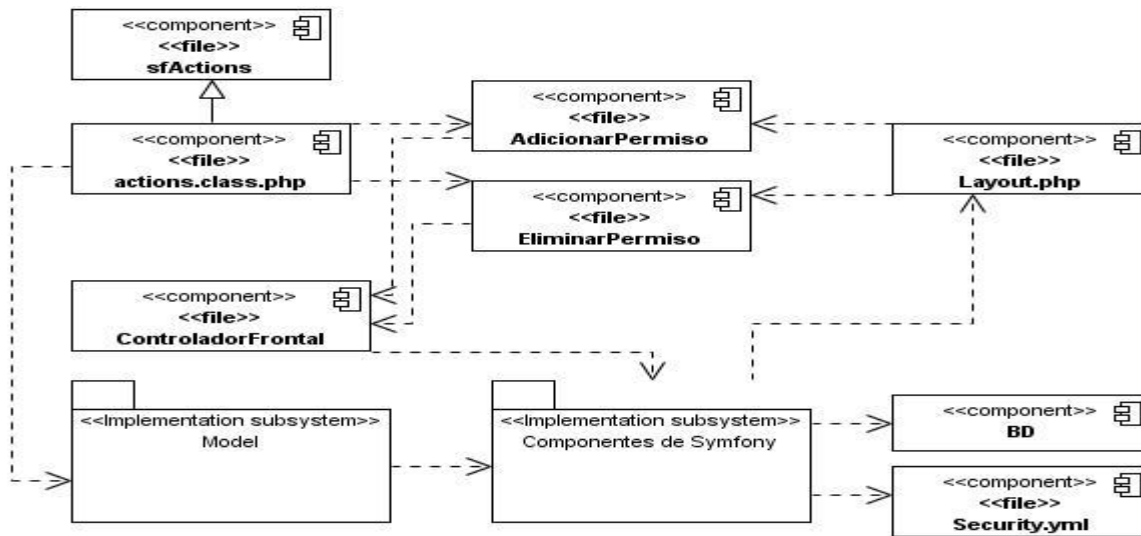


Diagrama de componentes CU Gestionar permiso.

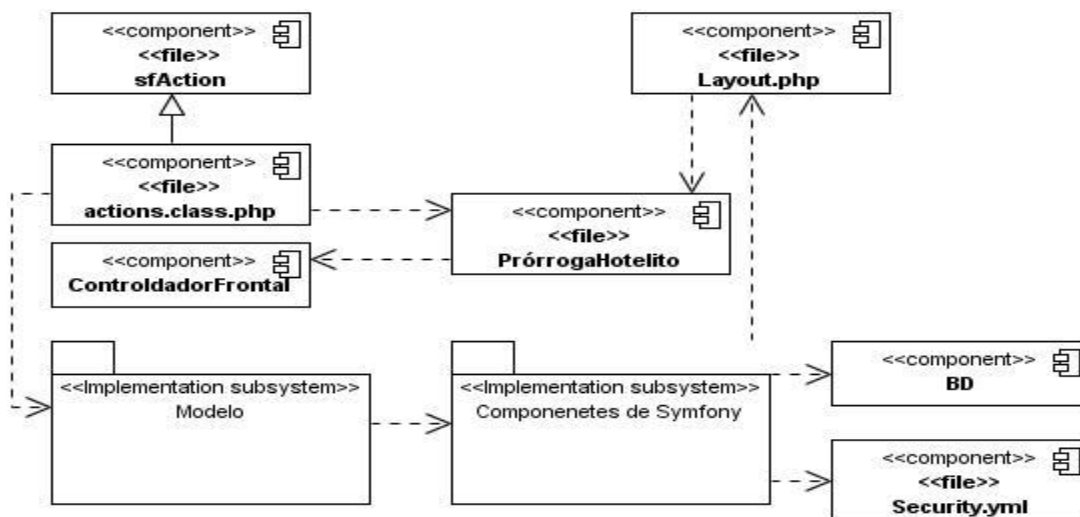


Diagrama de componentes CU Solicitar prórroga para visita del hotelito

Anexo #9 Operacionalización de las variables:

Variable Conceptual	Dimensión	Indicadores	UM
Sistema automatizado		Tiempo de desarrollo	Extenso
			Moderado
			Breve

	Factibilidad	Costo	Costoso Moderado Barato
		Esfuerzo	Alto Moderado Despreciable
Gestión de visitas al centro.	Optimización	Control	Bueno Malo
		Organización del trabajo	Bueno Malo
		Seguridad	Alta Media Baja
	Trabajadores	Agilidad en el proceso.	Rápido Medio Lento
		Satisfacción con el sistema.	Alta Media Baja
	Departamento de seguridad y Protección	Agilidad en el proceso.	Rápido Medio Lento
		Fiabilidad	Alta Baja Media
		Satisfacción con el sistema.	Alta Media Baja

	Directivos	Agilidad en el proceso.	Rápido Medio Lento
		Interacción con el sistema.	Sencillo Medio Complejo
		Satisfacción con el sistema.	Alta Media Baja